Final project for Math 390 Data Science at Queens College

May 24, 2020

By Pizon Shetu

Abstract

In this paper I will be predicting sale prices of houses in Queens, New York from 2016-2017, I will attempt to properly clean and analyze the data and fix any missing values. Afterwards use 3 algorithms which will be Linear Modeling, Regression Tree, and Random Forest to make a predictive model. We look to supplement any missing data by imputing onto it, run the algorithms and hopefully be able to utilize the model for useful knowledge and insight.

1. Introduction

For this model I seek to able to find what truly impacts house prices from all available data within this dataset, by looking into the nuisances of each variable and how much these variables affect the true price of a house. As mentioned earlier I will be using the following 3 algorithms, Linear Modeling, Regression Tree, and Random Forest. While each model has its own advantage, it is quite insight to see the varying results they each bring. To lightly touch on each:

Linear Model seeks to describe continuous variables as a function of predictor variables, upon which they can understand and predict the complexity of the data. Linear regression is used to create a Linear Model.

Regression Tree modeling allows for continuous or categorical variables as they use a decision to generate nodes which contain a test on a given input variable value. While the terminal nodes contain the predicted values for the output variable

Random Forest implements many decision trees while evolving our dataset. It uses a random sampling of training data while building its tree and a random subset of features when splitting its nodes. Out of the prior 2 Random Forest usually has more predictive indication when there is a large dataset.

2. The Data

Originally the dataset was comprised of 2,230 observation unfortunately we will only working with 528 since the rest do not have a sale_price for the residing rows. The dataset also contains 55 variables in other words columns from these variables we will be selecting our features, unfortunately again a lot of data is missing so I decided to drop all those that had a higher than 50% data missing or had more than 53 categorical response. In the end we were left with the following features:

approx_year_built	**cats_allowed**	**common_charges**
community_district_num	**coop_condo**	**dining_room_type**
dogs_allowed	**fuel_type**	**garage_exists**

kitchen_type **maintenance_cost** **num_bedrooms**

num_floors_in_building **num_full_bathrooms** **num_total_rooms**

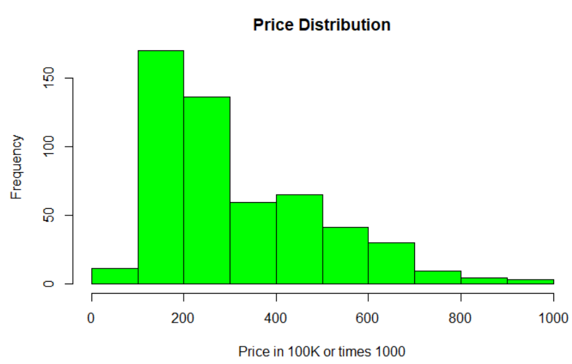
parking_charges **sale_price** **sq_footage**

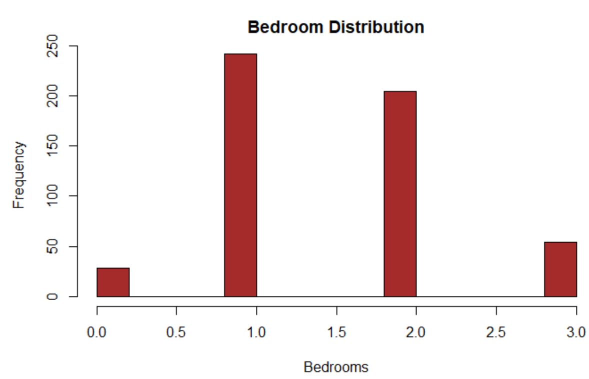
total_taxes **walk_score**

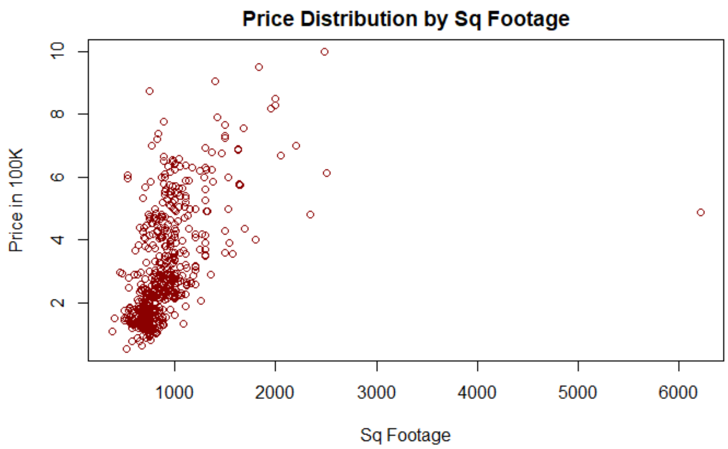
2.2. Featurization

In total there are 20 features that I have selected, all were provided from the raw while I decided to encode the yes and no response to binary response of 0 and 1, they were **cats_allowed, dogs_allowed, and garage_exists.** The categorical were left as it was but was factored and unordered for each of computer further down the line. The more impactful features to the eye were **approx_year_built, kitchen_type, maintenance_cost, num_bedrooms, num_floors_in_building, num_full_bathrooms, num_total_rooms, and sq_footage** some are continuous variables. The continuous features have the following means, standard deviation, and ranges, for **approx_year_built** the mean is 1962 and with a standard deviation (sd) of 20.5 and it ranges from 1915-to-2016. **maintenance_cost's** mean is $817.60 while it's sd is $352.90 and it's range is ($155-$4659). **num_bedrooms** mean is 2 sd is 1 and it ranges from 0-3**. num_floors_in_building** mean is 7 and it's sd is 6 and it ranges from 1-34**. num_full_bathrooms** mean is 1, sd is .5 and it ranges from 1-3**. num_total_rooms** mean is 4, sd is 1 and it ranges from 1-8 and **sq_footage** how much space the house takes up, its mean is 907.7 sq feet, its sd 366 sq feet, and it ranges from 375-6215 sq feet.

Here are few plots of different distributions of the data







2.3. Errors and Missingness

This dataset had a lot of missingness as far as errors there were a lot of the same response but in various string form between abbreviation and lowercase to uppercase for most I opted to make it binary response for example **garage_exists** had a lot of missingness and different form of answer but it seems no response were no thus I encoded all NA to 0 and else to 1. As far as other features missingness I used the package missForest to impute onto them

3. Modeling

My model seeks to perhaps give some insight into the true casual inputs of what yields a price for a house and reveal which factors really determine the value of a property. The 3 mentioned algorithms which will be used will Regression Tree, OLS Regression, and Random Forest. I had hope to perhaps utilize my model for future data prediction but unfortunately the results are not on my side as you'll see when I show my findings my model does subpar. But let us not be discouraged by results just yet and explore these algorithms and I hope the work I have done can be useful to someone.

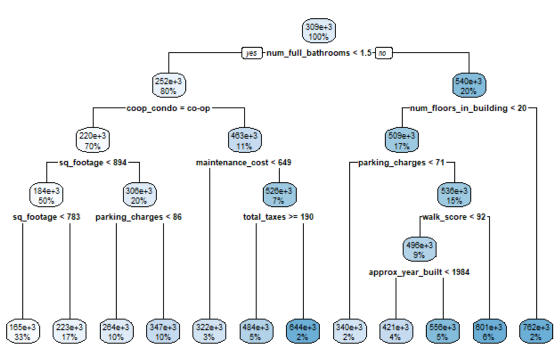
3.1 Regression Tree Modeling

For regression tree I was not able to utilize the YARF package as I was unable to install it properly so I opted to use the rpart package, with rpart it was quite simple to fit the training data and test on a the test data. I used RMSE as a error metric for this, while I did not yield great results I was able to get a $99K RMSE meaning our predictions are off +- $99k from the true sale_price now this is a scary I had originally found a set of features which lowered to $75K but not only was my model incompetent I felt as it was overfitting and even then it was way off from desirable numbers.

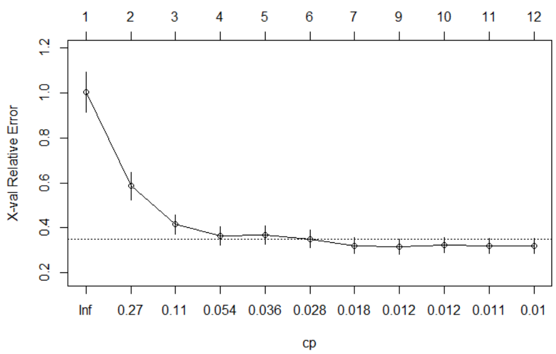
In the rpart package you are able to manipulate the two hyper-parameter min-split and max-depth essentially they control when the tree should split given the number of observation and the maximum of nodes it should have respectively. In order to find the optimal numbers for these two parameters I used a hyper-grid and ran a for-loop where it return the best min-split and max-depth which gave the lowest 'xerror' which is part of the cptable. The 'xerror' is related to PRESS statistics essentially it is the error on the observation from cross validation data.

After finding the optimal model I did further tweaking in hope of better RMSE and used 'bagging' with 10-fold CV to further improve my Regression Tree but to my disappoint we were only able to reduce RMSE to $85K out-of-sample and $75K in-sample.

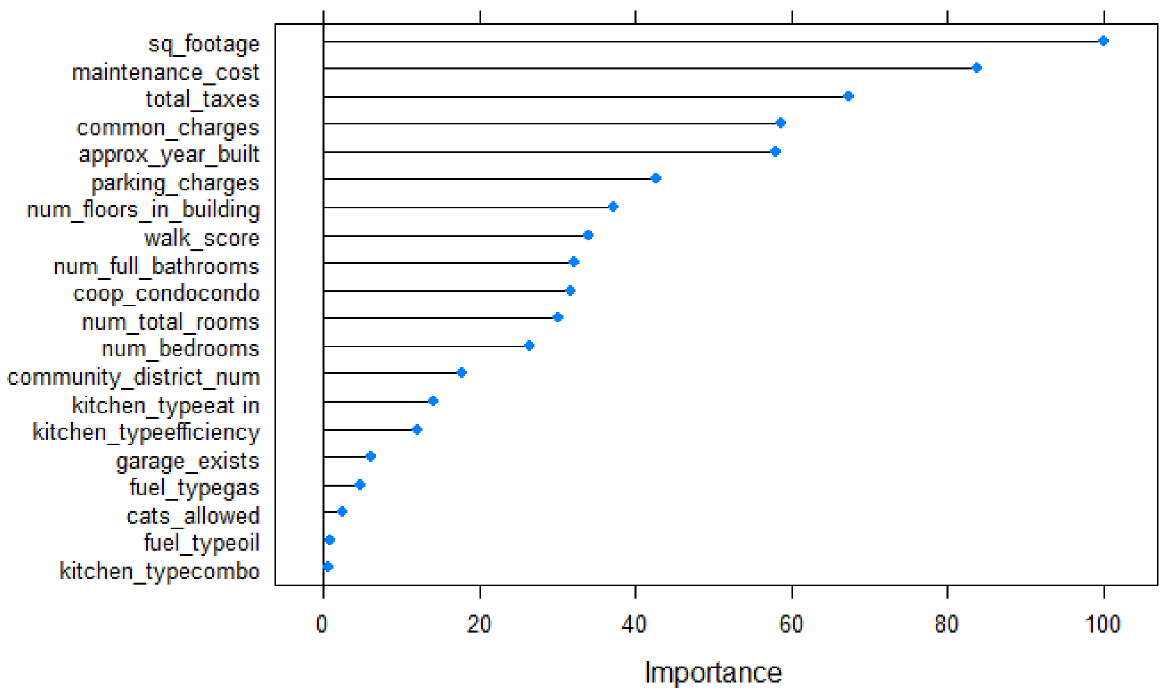
Here are some graphs to represent the tree and my variable importance levels after bagging



The Regression Tree



The error after a certain number of tree



The variable importance after bagging

It makes sense for sq_footage to be the biggest factor as the amount of land a house takes up has a large impact on its price. I am rather surprised at the expense variables to be so impactful such as maintenance, taxes, common and park charges then again more expensive housing usually relates to more affluent neighborhood thus costs are increased in those area’s but not sure how to take this as nearly 50% of all those data were imputed by missForest. Quite surprised to see the num_bedrooms be so low usually a house with more rooms yields a higher price. Walk_score is another one I did not expect to impact as highly as it did.

3.2 Linear Modeling

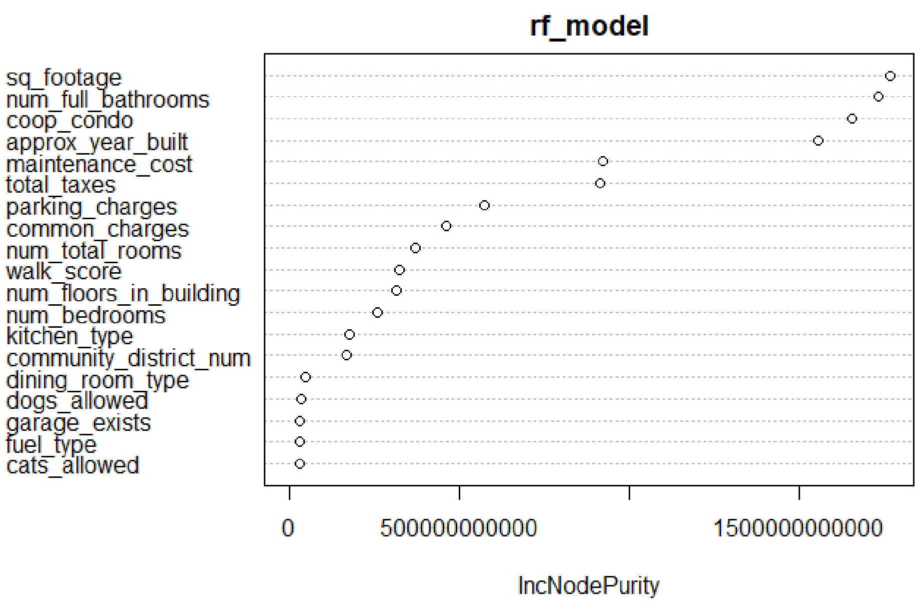
OLS model seem to do around the same as Regression Tree with an RMSE of $80K and an 80% R-squared, looking at the coefficients it seems coop_condo had the biggest impact followed by fuel_type, while coop_condo might make sense I believe fuel_type should not have as high impact compared how low of an impact sq_footage had. Also must note that num_fullbathrooms and num_bedrooms had a high impact on our y changes. I don't believe OLS would be a ideal algorithm to predict housing prices due to the complexity of how each feature interact and as we can see from the coefficient estimates certain features played a bigger role than those who are more deserving in the real world for example.

```
Call:
lm(formula = sale_price ~ ., data = house_imp)

Residuals:
    Min      1Q  Median      3Q     Max 
-379438  -46108     146   41249  345172 

Coefficients:
                           Estimate  Std. Error t value             Pr(>|t|)
(Intercept)            -128449.5435 607200.9045  -0.212               0.8325
approx_year_built          -28.2861    304.7232  -0.093               0.9261
cats_allowed             23751.0179  10066.4099   2.359               0.0187
common_charges              83.4643     41.2029   2.026               0.0433
community_district_num    2127.5905   1281.8243   1.660               0.0976
coop_condocondo         199876.3376  13524.3052  14.779 < 0.0000000000000002
dining_room_typeformal   25399.5743   9445.4113   2.689               0.0074
dining_room_typeother     2137.7691  12230.2467   0.175               0.8613
dogs_allowed             -6599.2462  11241.9137  -0.587               0.5575
fuel_typegas              5775.1591  26628.5103   0.217               0.8284
fuel_typeoil             14446.8574  27475.1750   0.526               0.5993
fuel_typeother           16892.4149  36660.6692   0.461               0.6452
fuel_typeOther          115196.8500  86860.3889   1.326               0.1854
garage_exists            -2907.9905  10042.3711  -0.290               0.7723
kitchen_typecombo       -60595.6892  83921.8570  -0.722               0.4706
kitchen_typeCombo       -37879.8738  83342.1781  -0.455               0.6497
kitchen_typeeat in      -47701.3563  82488.9443  -0.578               0.5633
kitchen_typeEat in       39242.3861 101873.4874   0.385               0.7002
kitchen_typeEat In      -68455.9238  84857.2335  -0.807               0.4202
kitchen_typeefficiency  -78734.2236  82601.6006  -0.953               0.3410
maintenance_cost           100.5978     18.4481   5.453        0.00000007804
num_bedrooms             51056.1725   8717.6077   5.857        0.00000000858
num_floors_in_building    3778.0491    832.7677   4.537        0.00000716482
num_full_bathrooms       56431.2601  12724.3149   4.435        0.00001133871
num_total_rooms           6401.3981   5817.7324   1.100               0.2717
parking_charges            916.7265    104.0300   8.812 < 0.0000000000000002
sq_footage                  15.9451     14.4349   1.105               0.2699
total_taxes                  0.1203      4.1319   0.029               0.9768
walk_score                   6.6691    317.9400   0.021               0.9833
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 80610 on 499 degrees of freedom
Multiple R-squared:  0.8091,	Adjusted R-squared:  0.7984 
F-statistic: 75.53 on 28 and 499 DF,  p-value: < 0.00000000000000022
```

3.3 Random Forest

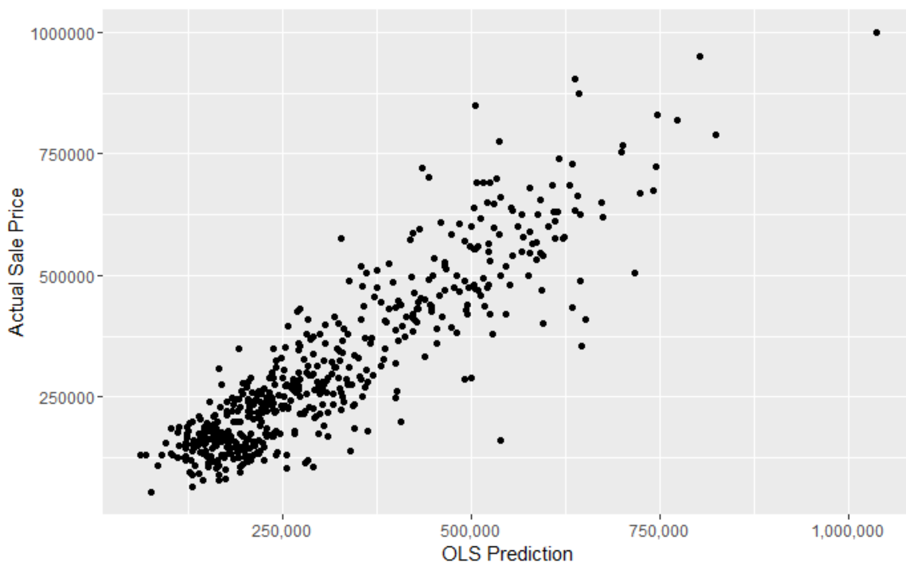
Random is widely used, because of its simplicity and variety as it can be used for both classification and regression. It is adaptable, and simple to use algorithm, even deprived of hyper-parameter tweaking, it will produce great results. I used randomForest package to run my algorithm where it yield a RMSE of $77K out-of-sample and a 50K in-sample, while this is significantly better than our prior results with the other two algorithms I would be lying if I didn't say I failed to capture the full picture of this project and failed to maximize the best features to yield the greatest results.

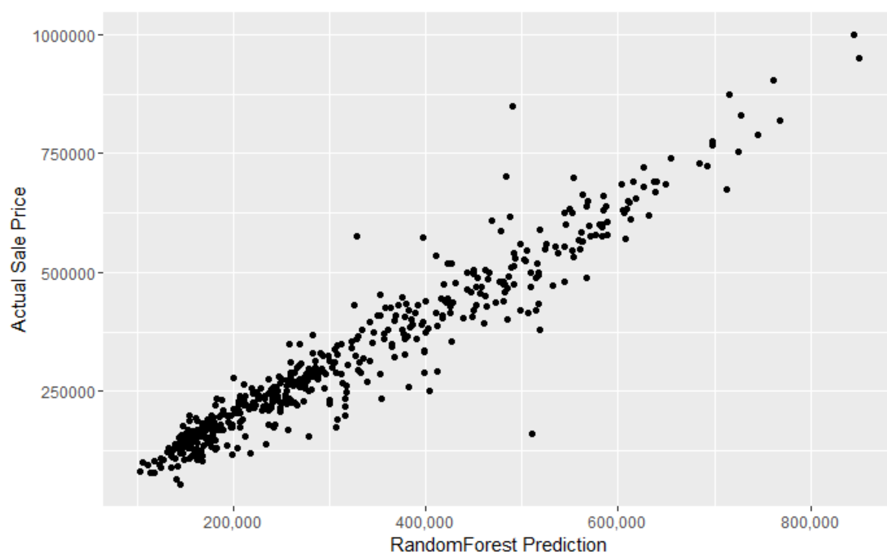
Here is a plot of variable importance for Random Forest

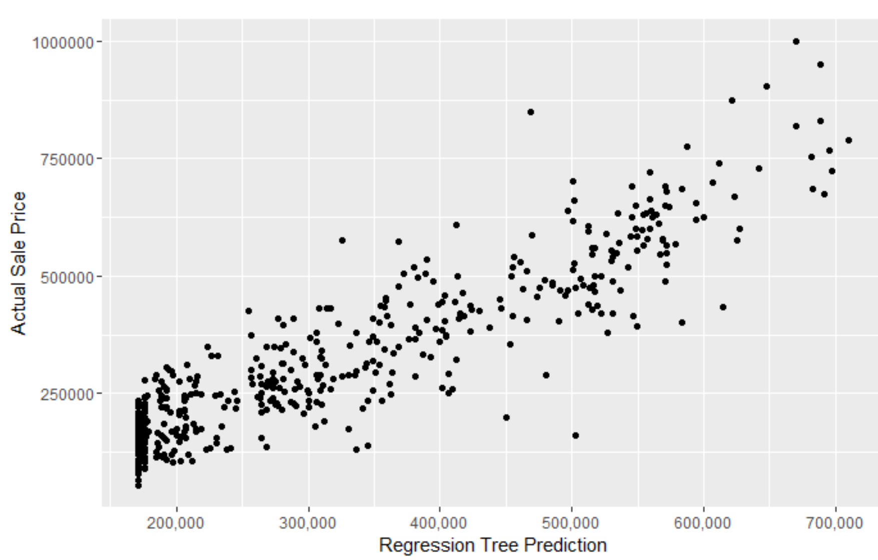


4. Performance

While it was not performance we hoped it was the best I did, out of all the algorithms Random Forest yielded the best results special for in-sample, nevertheless our model did outperform the null model as the RMSE for our model yielded 50K and 75K for in-sample and out-of-sample whereas the null-model yield a RMSE of $147K. For more of a comparison I did an in-sample comparison of all 3 algorithms predicted price vs actual price. You can imagine the following as if the predictions were perfect you would have a straight linear line where x point = y point and as you can see our Regression Tree did the worse and Random Forest did the best


1000000
750000
500000
250000
Actual Sale Price
250,000
500,000
750,000
1,000,000
OLS Prediction


1000000
750000
500000
250000
Actual Sale Price
200,000
400,000
600,000
800,000
RandomForest Prediction


1000000
750000
500000
250000
Actual Sale Price
200,000
300,000
400,000
500,000
600,000
700,000
Regression Tree Prediction

5. Discussion

We had hoped to make models which we could utilize in real-world predictions unfortunately in my eyes I feel the models came up a bit short, I suspect this is due to poor data mining and cleaning and perhaps need better set of features, nevertheless it did beat the null model. In future endeavors I would have taken a different approach and perhaps used a log function on sale price and tried other various methods that perhaps might yielded better results. Also, I would have opted for better mining and cleaning of the data methods as I believe this was a major contributor to the poor results.

I would have like to see much larger usable observations as most were negligible or missing. I would have also like to have seen more variables such as money spend on renovation, rent price for the house either by room or by floor as most houses are rented out by those two forms.

Regression Tree did the worst out of the 3 models I suspect this is due to the complexity of our models, I suspect increasing the complexity will yield to a greater Regression Tree results, and perhaps even our OLS.