**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**

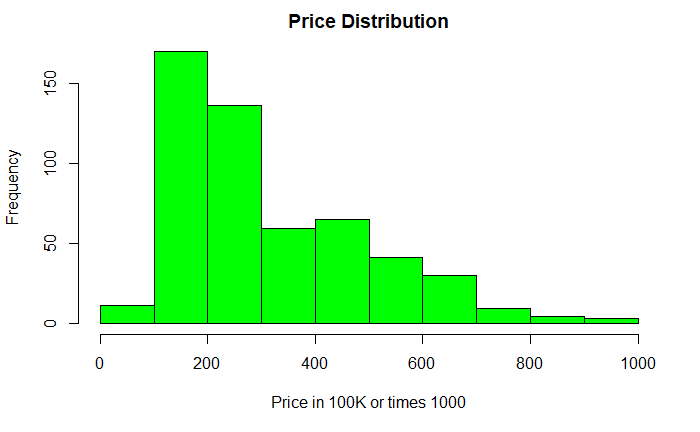
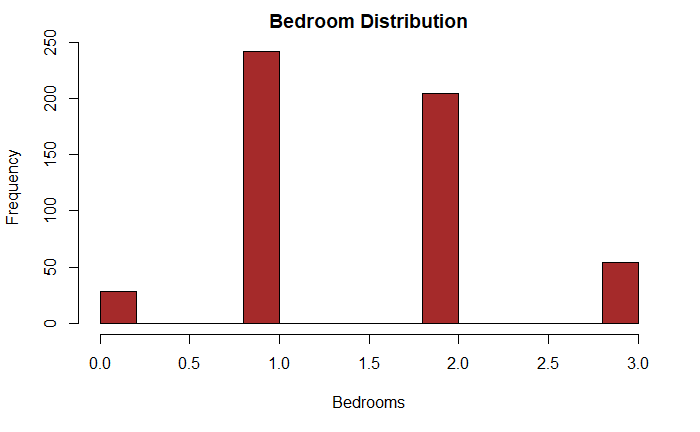
**parking\_charges sale\_price sq\_footage**

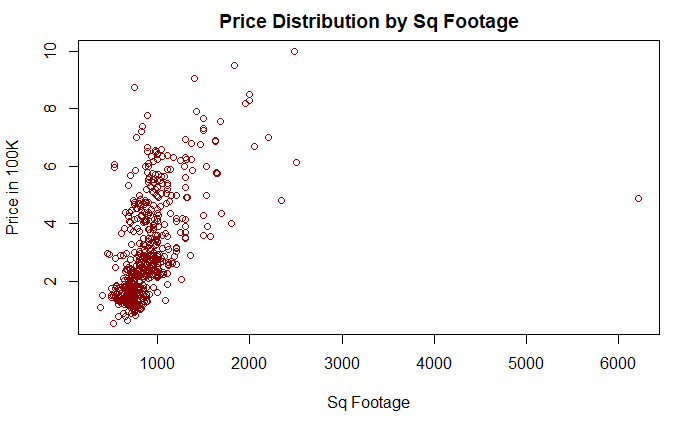
**total\_taxes walk\_score**

**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.

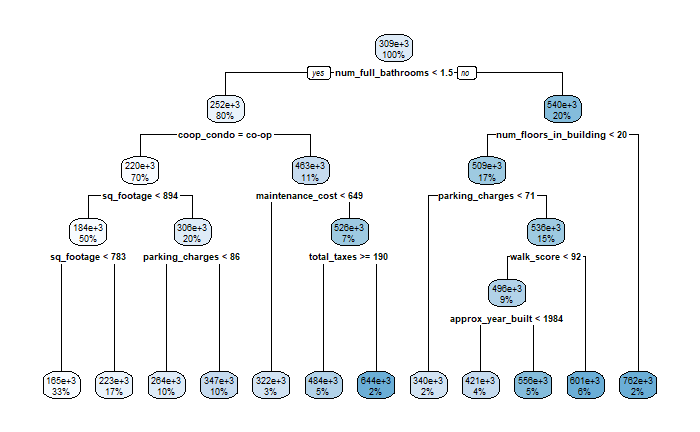
**3.1 Regression Tree Modeling**

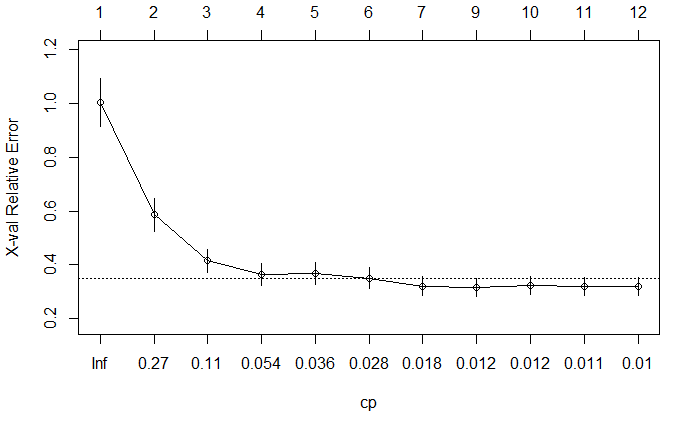
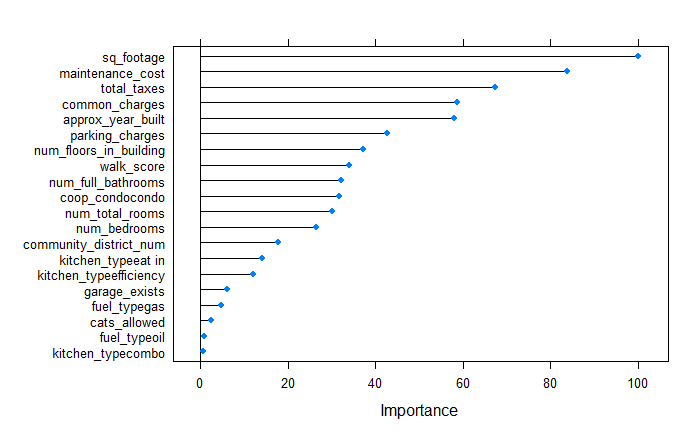
For regression tree I was not able to utilize the YARF package as I was unable to install it properly so I opted to use the rpart package, with rpart it was quite simple to fit the training data and test on a the test data. I used RMSE as a error metric for this, while I did not yield great results I 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**

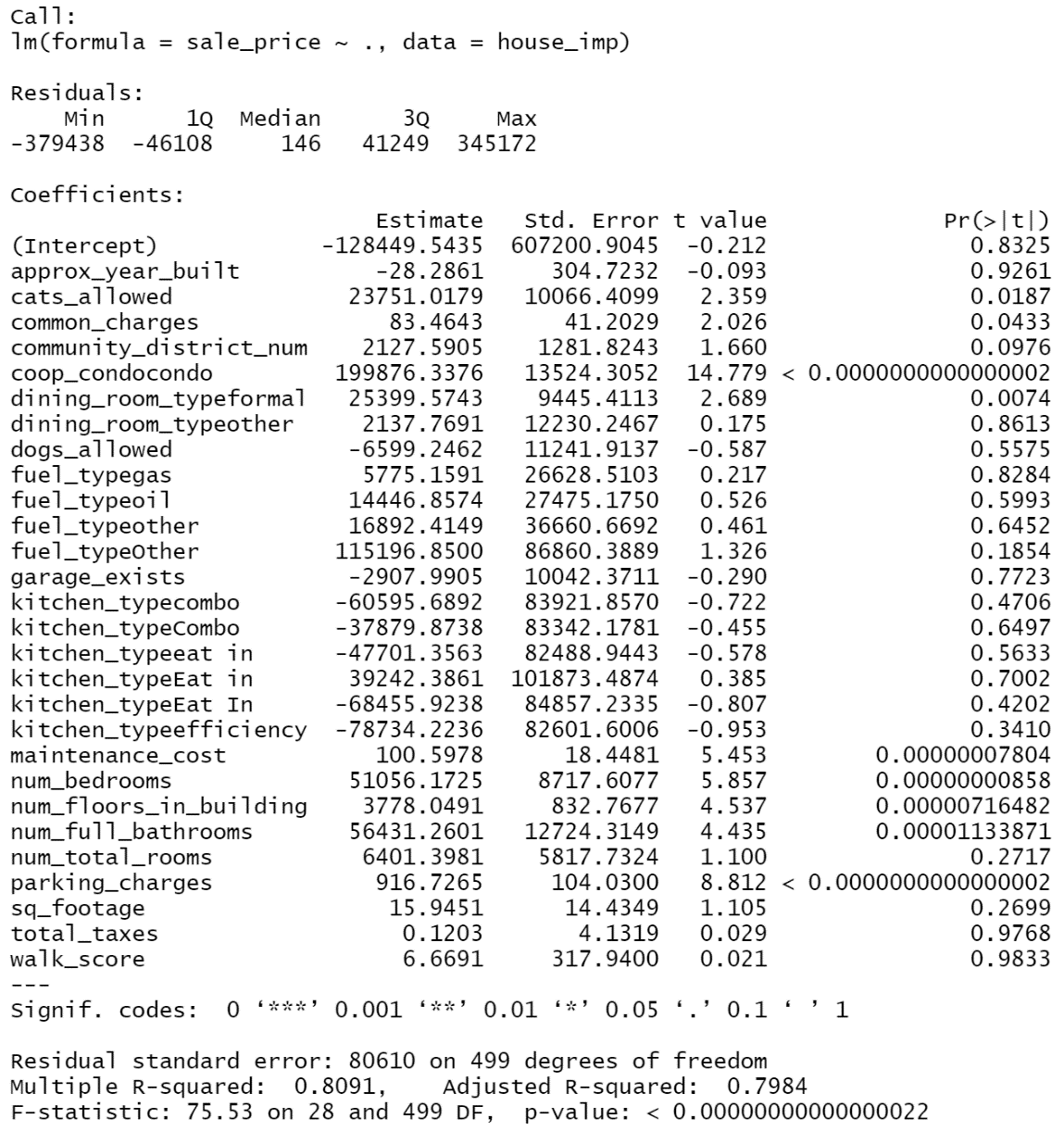
**The error after a certain number of tree**

**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**

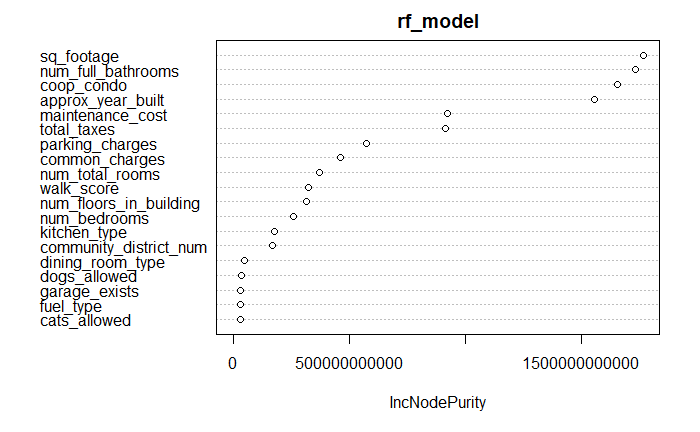
OLS model seem to do around the same as Regression Tree with an RMSE of $80K and an 80% R-squared, looking at the coefficients it seems coop\_condo had the biggest impact followed by fuel\_type, while coop\_condo might make sense I believe fuel\_type should not have as high impact compared how low of 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.



**3.3 Random Forest**

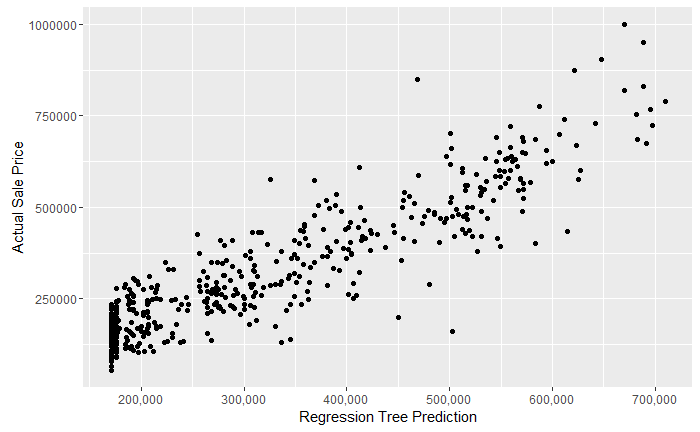
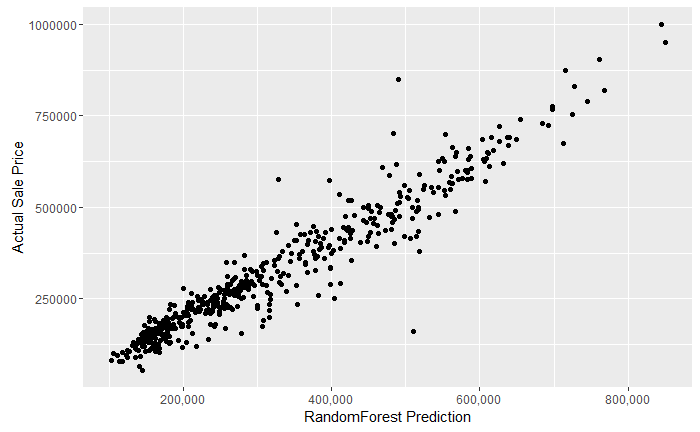
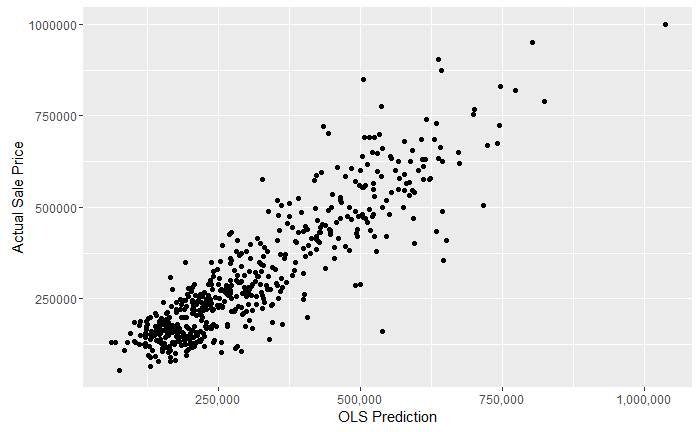
Random is widely used, because of its simplicity and variety as it can be used for both classification and regression. It is adaptable, and simple to use algorithm, even deprived of hyper-parameter tweaking, it will produce great results. I used randomForest package to run my algorithm where it yield 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



**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.