Queens Apartment Price Prediction

Final project for Math 390 Data Science at Queens College

May 24, 2020

By: Tyron Samaroo

Abstract- Predicting the apartment price phenomenon seems like a simple task given the amount of data that can be collected. This is solved by finding those significant features that can truly estimate the predictive outcome. Once these features have been found. there is a need to correct any errors and handle any missingness. After doing this, different models like Random Forest, Regression Trees, and Linear Model(OLS) are used and fine tuned to provide a useful model that can perform accurate real life predictions.

I. INTRODUCTION

The phenomenon of apartment sales price will be the response variable which we will call y. One problem we have when predicting a phenomenon is that we do not know the true causal inputs $(z_1, z_2, ..., z_p)$ and the true relationship t between them. We can express this as a formula below.

$$y = t(z_1, z_2, ..., z_p)$$

The next best thing we can do is approximate the $(z_1, z_2, ..., z_t)$ with measurable data which we can call $(x_1, x_2, ..., x_p)$ and the relationship between them f. Now the phenomenon can be expressed as follows.

$$y \approx f(x_1, x_2, ..., x_p)$$

$$y = f(x_1, x_2, ..., x_p) + \delta$$

We can only approximate the phenomenon so we can include δ to be the error due to ignorance which we have no control over. This will be the predictive model.

A predictive model's job is to perform real world approximation on a given reality or a phenomenon. It is a model that is trying to find meaning to the phenomenon it is tasked with, by finding meaningful observable features. In this case predicting Queens apartment prices is the phenomenon that is being predicted based on observable features $(x_1, x_2, ..., x_p)$ that is provided from the 2016 and 2017 apartment sales data in Queens, New York. But be wary, a famous statistician George Box once stated that "All models are wrong but some are useful." (Box, 1987). Geroge is warning many to not expect much from models, they only provide a way for humans to have a better understanding of a problem that is out of the scope of their knowledge.

In order to solve this problem three different modeling techniques such as linear modeling, regression trees modeling, and random forest

modeling are used. Linear Model is tasked with finding meaning between each of the observed features and its influence on the apartment sales price. Regression Tree Model tries to find the best possible way to split the dataset D and construct a binary decision tree based on the given split. Random Forest extends bagging that performs a bootstrap sampling of the data. Random Forest Model computes many regression trees. While it computes these trees at each of their node construction, it splits on a subset of features in order to reduce correlation which also reduces variance.

II. THE DATA

The data used is Queens, New York housing data from 2016 to 2017 collected from MLSI. This dataset contains information from mainland Queens.

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

Figure 1. Zip codes from houses in the dataset

The dataset contains 2,230 numbers of rows and 55 columns. The rows represent the number of observations collected and the columns represent features about each observation. Using the skim() method from Skimr R Package, 36 factor variables, 6 logical variables, and 14 numeric variables were found. There is also a lot of data that is missing especially for sales price, our response y. Many of the observations can be concluded to be unnecessary "junk" that is not representative in predicting sales price. After evaluating the data, the number of columns/features was reduced to 15 that are believed to provide a meaningful prediction. The features includes categorical variables such as (fuel_type, coop_condo, dining_room_type, dogs_allowed, cats_allowed, garage_exist) and continuous variables such as (sq_footage, walk_score, common_charges, num_full_bathrooms, num_floors_in_building, num_full_bathrooms, maintenance_cost, num_total_rooms, num_bedrooms, appox_year_built)

Table 1: Data summary

Name	df
Number of rows	2230
Number of columns	55
Column type frequency:	
factor	36
logical	5
numeric	14
Group variables	None

Figure 2. Original Data Summary from Skimr.

III. FEATURIZATION

The original dataset looks like it has plenty of observations with many features but a closer look at the data shows that the *sale_price* that is missing 1702 observations, that's about 76% of our data missing its response *sale_price.* All observations with missing *sale_price* were completely dropped, leaving a total of 528 observations. When observing the 55 features availability, many of them don't really help with explaining *sale_price.* Features such as (Title, Description, CreationTime, WorkerId, Expiration) were removed. On the other hand, features like *sq_footage* were kept because they represent the size of the land. The average square footage was around 965, while

the majority of the square footage was under 3,000 except for one that was 6,000. Since this was the only occurrence, it was treated as an outlier and was removed from the data. Figure 3 is a scatter plot that visualizes sales price and square footage where you can see the distribution of the data. It ranges from 375 square feet to 6,215 square feet and has a standard deviation of about 364 square feet. Clearly this provides some insight about sales price, as some sort of correlation can be seen.

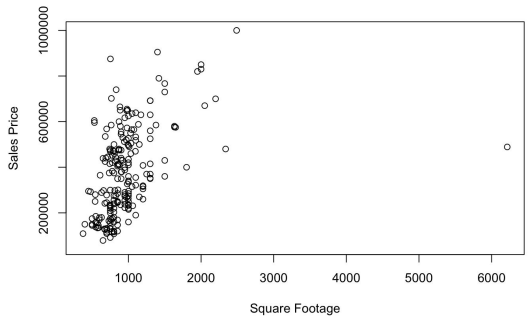

Figure 3. Sales price according to Square Footage

Another feature *approx_year_built* ranged from being built in 1915 to 2016 with a standard deviation of 20 years. There isn't a clear visual relationship as when compared to sales price (Figure 4). It seems like there is almost no correlation, but there is at least something that the model can learn from.

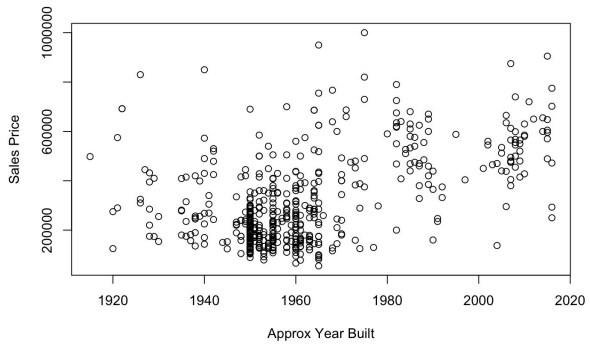

Figure 4. Sales price according to Approx Year Built

Another interesting feature was coop_condo; it is a categorical variable that has about 76% co-ops and 24% condos. This feature basically classifies an apartment as being part of a cooperative ownership that's owned by multiple people or condo that is individually owned.

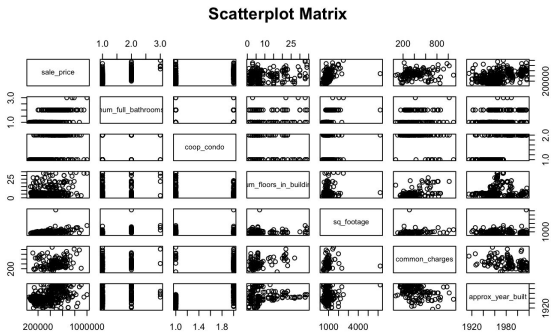

Figure 5. Scatterplot Matrix of some of the features used

IV. ERROR AND MISSINGNESS

While exploring the dataset, there were a couple of errors. For the categorical feature *cats_allowed* there were some *y* labels with the existing *yes* and *no* labels. It did not make sense to have *yes* and *y* since it makes sense to have only two choices when asked if cats are allowed. So all observations that have a *y* label were changed to *yes* label. Another categorical feature *fuel_type* also had two levels *other* and *Other*, it seems like it was a mistake so the two factors were combined to just be *other.* Another issue was *common_charges* and *maintance_cost*, they were both factors but were changed to numeric as it makes more sense since these are continuous variables. Lastly the response *y* which is the *sales_price* was also a factor that was also changed to numeric type.

From the beginning, there was a huge problem with the given dataset. There are 2,230 observations with their 55 features. The problem is that for the response *sales_price* there were a total of 1702 observations missing their sales price, that's about 76% of the data missing the feature that will be predicted on. Given that *sales_price* is being predicted it will not make sense to impute it. Thus it was removed from the dataset which was now left with 528 observations. After doing some featurization that was mentioned beforehand a total of 15 features were left to make a prediction on *y*. There was still some more missingness in *dining_room_type, sq_footage, common_charges, maintence_cost* which were imputed using two methods, missingness vector and missForest. Data that was missing was encoded in a new feature as 0 or 1 depending on if it was missing or not. Then the data that was missing was imputed using missForest that uses the random forest algorithm to impute all missing values. Now the dataset has no missing values and now has 22 features because missingness vector was used.

V. MODELING

The data is now fully prepared for modeling. There are many different types of models that can be used in the machine learning world. We will only discuss three different types, a linear model, regression tree model, and random forest model. All three use different approaches to make accurate predictions. Each with different statistical interpretation and real life interpretation.

VI. REGRESSION MODELING TREE

Regression trees use given data to build a binary tree that partitions the data as best as it can at each given split. It is an iterative process that keeps on splitting on each interaction into a left node and a right node. At each node level, a computational rule is used to split the data using a given feature and value and keeps on doing it until it cannot split anymore. Regression trees are one of the most interpretable models as you can visualize what features it splits on. It's important to note that it uses a greedy algorithm and splits on some local optimal feature which may or may not be the global optimal feature. It only considers the next possible split as it makes it way down the tree and not the overall best possible split.

When a regression tree was applied to the housing dataset it returned multiple important features that it found to be important interactions between each feature that best encapsulates sales price. Figure 6 shows a graphical representation of the features the regression tree algorithm splits on. The most significant split that it first found was the number of full bathrooms that was less than 1.5. This perhaps makes sense since usually luxurious houses have more than one bathroom. Next it splits on co-op and condos with a number of full bathrooms less than 1.5. A closer look at the data shows that 76% of the observations are co-ops which makes sense since apartments in co-ops usually aren't that spacious and could only fit one bathroom. Then on observations that are co-op, it split on square footage that's less than 811. Square footage of an apartment makes perfect sense

since the more space you have, the more things you can fit on that land hence could mean an increase in price. Down this line it splits again on square footage. Now back to the top with full bathrooms greater than 1.5, it splits on the number of floors in the building. If there are a lot of bathrooms it's safe to assume that the apartment might have many floors. Then it splits on walk score; walk score is rating the walkability to public transportation. Higher value apartments that are closer to resources like public transportation does make sense to be priced more heavily. The next split uses square footage which is no surprise as this is an obvious idea that drives housing prices. This continues onward as you can see it's very interpretable the longer you think about it.

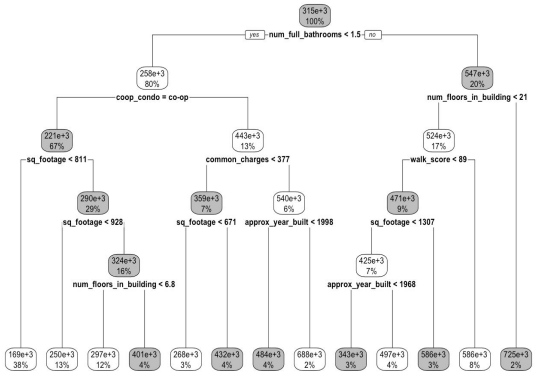

Figure 6. Graphical representation of Regression Tree

To train the regression tree 90% of the data was used as a training set and 10% was used as a testing set. The regression tree will only have access to the training set to best predict the sale price with the given features. After training the model it did really well in the sample. It reported an R-squared of 98.8% and an RMSE of 19,320. This seems pretty good but when calculating out of sample using the testing set the regression tree hasn't seen yet, it reports an R-squared of 79.6% and a RSME of 87413. This means it truly did 79.6% better than just taking the average price to make a prediction. This also hints that there was overfitting on the training data, so it failed to generalize correctly for future prediction. It is also questionable if this is a model to be trusted too closely since its prediction is within ±$87,413.

VI. LINEAR MODELING

Next Ordinary Least Square is used to model the data. Same concept of splitting the data is used as in the decision tree, the only thing that is changed is the algorithm. After training the model, the in sample reports an R-Squared of 76.5% and RMSE of 88710 is reported. The out of sample metric that was computed using the training set reported an R-Squared of 80.4% and an RMSE of 85564. Here we see an inverse result compared to the regression tree model. Even though the in sample RMSE and R-squared was lower it still did better out of sample. It not only did better out of sample for linear model but also better than the regression tree model. But it is still not an overwhelmingly better model since the prediction is within ±$85,564.

When comparing mutually observed features A and B sampled in the same fashion as features in the dataset, when A has an observation x value one unit larger than B's x measurement but both A and B share same measurements $x_1, x_2, ..., x_p$ then A is predicted to have a response y that differs by some unit

b on average from response *y* of *B* assuming the linear model is true. In other words, the coefficients in the model show how much each feature is offset from the response variable sale price when the other features are held constant. Looking closely, the most impactful coefficient comes from the coop condo feature that is followed by the number of full bathrooms, and then by the number of bedrooms. This makes OLS just as interpretable as regression trees, as there are similar features that highly impact our response *y*. Given that this is a regression task, a linear model like OLS shouldn't have much issues with making predictions on sale prices given reliable features that approximate the phenomenon *y* closely. In this example OLS seems like a better pick than a regression tree since it outperforms it out of sample with both R-squared and RMSE.

```
Call:
lm(formula = sale_price ~ ., data = data.frame(df_train), subset = set.seed(28))

Residuals:
    Min      1Q  Median      3Q     Max
-353328  -46131   -1660   43358  353257

Coefficients:
                                 Estimate Std. Error t value             Pr(>|t|)
(Intercept)                    -298395.10  659717.40  -0.452             0.651266
cats_allowedyes                   9080.70   11575.97   0.784             0.433193
common_charges                      31.45      52.67   0.597             0.550712
coop_condocondo                 268580.86   25419.10  10.566 < 0.0000000000000002 ***
dining_room_typedining area      23700.90   89632.03   0.264             0.791574
dining_room_typeformal            7681.67   11135.26   0.690             0.490644
dining_room_typeother            32171.80   14691.74   2.190             0.029053 *
dogs_allowedyes                  19982.52   12690.44   1.575             0.116050
fuel_typegas                     -3832.21   29743.63  -0.129             0.897541
fuel_typeoil                     -4827.69   30620.42  -0.158             0.874794
fuel_typeother                    5305.87   33823.15   0.157             0.875417
garage_exists                      802.17   12224.26   0.066             0.947709
maintenance_cost                   121.29      26.87   4.514        0.0000081193 ***
approx_year_built                   33.66     336.42   0.100             0.920351
num_bedrooms                     54268.57   10137.47   5.353        0.0000001381 ***
num_floors_in_building            4939.70     918.37   5.379        0.0000001209 ***
num_full_bathrooms               82887.84   14540.51   5.700        0.0000000217 ***
num_total_rooms                    378.61    6557.42   0.058             0.953983
sq_footage                          23.79      16.32   1.458             0.145619
walk_score                        1134.48     336.43   3.372             0.000811 ***
is_missing_common_charges        42305.48   25406.07   1.665             0.096576 .
is_missing_dining_room_type       7221.66   10162.44   0.711             0.477687
is_missing_maintenance_cost     -26899.32   22966.64  -1.171             0.242126
is_missing_approx_year_built     23817.49   37540.65   0.634             0.526113
is_missing_num_floors_in_building  14964.93   10575.50   1.415             0.157745
is_missing_sq_footage            -15340.57    8925.11  -1.719             0.086338 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 88710 on 449 degrees of freedom
Multiple R-squared:  0.765,     Adjusted R-squared:  0.7519
F-statistic: 58.46 on 25 and 449 DF,  p-value: < 0.00000000000000022
```

Figure 7. Summary of OLS output table

VII. RANDOM FOREST MODELING

Random forest is another algorithm that has amazing predictive power. As the name implies, it is a “forest” that is filled with regression trees. Each tree in the random forest outputs a prediction and the best is picked. Random forest uses methods such as bagging that samples with replacement the dataset to fit the best model. Then, during each of its node construction it splits on a subset of features to build each tree. This makes each tree different hence they are uncorrelated, which in turn reduces variance. Random forest is also non-parametric. There is so much to gain from using it as it significantly reduces bias and variance while providing free validation while fitting the model. Random forest makes sure it gets as much information as it can get from the given data set to make the best prediction it can. The only con to random forest is that it is not that interpretable like regression tree models or linear models.

VIII. RANDOM FOREST RESULT

The same train and test split of the data is used here again but to validate the random forest model. After training the model the OOB R-squared reported was 80% and OOB RSME was 78439. This is calculated with the observations that are left out when sampling for each tree. On average if the number of trees that random forest is trained on is large enough the oob validation set is $1/e$ that is about ⅓ the number of models. Looking at OOB it leads to believe that it will do 80%

better than taking the average sale price with its prediction within ±$78,439.

Next when evaluating how the model did in sample an R-squared of 96.6% and RMSE of 32928 was reported. However when calculating the out of sample R-squared and RMSE the values 82% and 82058 were reported respectively. Thus, on the test set that was held out, random forest did better than both regression tree and OLS. There is 82% improvement over taking the average price for prediction within a range of ±$82,058. Even though we got a better R-squared and RMSE, it was off by ±$82,058 which is a significant difference.

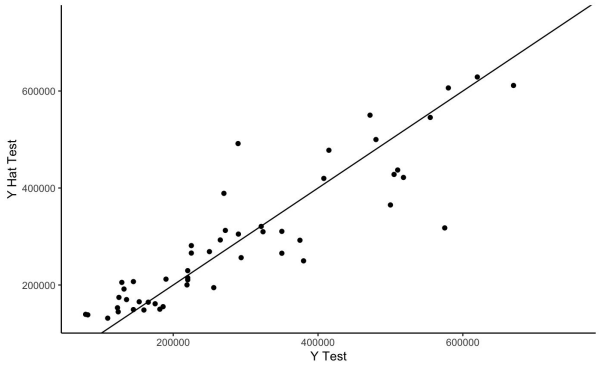

Figure 8. Shows how astray prediction is from test set

IX. DISCUSSION

The phenomenon of apartment sales price which is the response *y* is predicted using a combination of features in Queens apartment dataset. One the biggest issue with this dataset is that 1702 or 76% of the data was missing its response *y*. The most honest thing to do what many may not do is remove those data points. It's nonsensical to try to predict for a response you don't know for a model that is being built to predict the response.

Next, there were many features available but many were not related to sale prices at all and were removed. Once features that were determined to be of no help to predict the phenomenon were taken care of, all that was left was to fix errors and missinginess in the dataset. Errors such as variables that should be continuous were changed to be so. The remaining missingness of the data was imputed using missForest and adding a binary vector indicating what was missing to the existing data. It might be questionable why adding a vector that indicates what was missing but doing so might be helpful. There may be an underlying reason why that data is missing which may lead the models to learn something that many wouldn't be able to see.

Having so little data to use to build the model is one the reasons why this model would not be ready for production. The best model so far is the random forest model that reported a R-squared 82% and RMSE of 82058. With RSME being so high, it might not be a good idea to offer someone a prediction where you can potentially be off by ±$82,058. If more data is collected, the model would be drastically improved as it has enough features that can approximate the phenomenon as close as possible to its true self.