Econ 484 Final

CJ Robinson

6/10/2020

Introduction

Housing prices are used by various professionals and academics to research trends in the market and provide helpful information for buyers or sellers. The factors that may go into a person's decision to set sell their house at a certain price are varied and oftentimes unobservable. Even still, predicting housing prices using econometrics and machine learning can provide actionable insights or attempt to anticipate future trends.

In the following sections, I develop a model which predicts housing prices in each country. I also use an interpretable model to find that there are several significant drivers of housing prices in Peru and Ecuador, although they are distinct.

Data Description – Task 1

The data from this report comes from Kaggle.com, a website with a data repository for machine learning problems. The data focuses on property listings in two countries, Peru and Ecuador, from March 2019 to March 2020. Several variables are in Spanish, but for the purposes of this paper will be translated into English. Variables include price, bedrooms, bathrooms, surface coverage, location, latitude, longitude, type of advertisement, several important dates related to the listing, and several textual variables like title and description.

I made several decisions regarding the data cleaning and feasibility of each variable. The number of NA's were too high for several variables, including "l3"-"l6" which represent more granular locations, so they were not included in the analysis. "l1" was simply the country and also not included. Additionally, the variable "rooms" was not included as it was not helpful for many observations and was made up of mostly NA's.

"start_date" and "created_on" were highly colinear, so I choose to only include "start_date" in my analysis. To process these dates, I subtracted each from 01-01-2019 to get the number of day since the beginning of 2019. "end_date" had many observations without end dates that were coded as the year 9999, so this column was not included. Additionally, the variable "ad_type" had only one level, so it was not included. "id" was not pertinent to housing prices, so it was also not included. "surface_total" and "surface_covered" were similar variables, and there were too many NA's in "surface_covered" so I only used "surface_toal." I hot-coded two variables, "currency" and "price_period" since the values were either NA or another value. Currency ended up only being applicable with Peru data as it was the only one with variation in currency once other data cleaning had occurred, and neither dataset had any variation in price_period once I had done the data cleaning. Finally, I did not use textual variables like description and title. This left the variables "price", "lat", "lon", "bedrooms", "bathrooms", "surface_total", "property_type", "operation_type", and "currency" (only for Peru).[1]

Below are summary statistics for each country's training set:

Table 1: Peru Training Set

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
start_date	27,682	218.294	103.422	60	130	288	448
lat	27,682	−12.109	2.097	−18.116	−12.133	−12.071	−3.511
lon	27,682	−76.620	1.931	−81.273	−77.053	−76.962	−69.200
bedrooms	27,682	3.448	2.212	0	3	4	45
bathrooms	27,682	2.846	1.638	1	2	3	20
surface_total	27,682	294.338	4,416.244	10	90	223	320,000
price	27,682	347,533.800	1,230,099.000	50	50,000	335,000	63,050,000

Table 2: Ecuador Training Set

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
start_date	43,008	192.619	68.327	60	136	236	311
lat	43,008	−1.426	1.165	−4.343	−2.192	−0.188	1.061
lon	43,008	−79.197	0.744	−90.431	−79.898	−78.484	−76.857
bedrooms	43,008	3.043	1.527	1	2	3	30
bathrooms	43,008	2.810	1.440	1	2	3	20
surface_total	43,008	229.135	1,040.697	10	91	209	110,000
price	43,008	112,112.800	203,533.300	50	700	147,170.8	14,000,000

[1] All code for data cleaning is in appendix

Methodology - Task 1, 3 & some of Task 4 (Model Selection/Feature Selection)

Building a Predictive Model - Boosting

There are several methods available to build a predictive model, but I ultimately chose to use Boosted Regression. Boosting provides many of the advantages of tree-based methods while slowing down the learning process. Its basis is similar to that of random forests and bagging in that it divides that data along different independent variables, forming a decision tree. Each split is considered to be a node, and at the end of each of these tree branches are terminal nodes. Using the training data, data is split into each of these terminal nodes and then averaged to create predictions for new data. Single trees without cross validation or pruning can lead to overfitting of the training data depending on the tuning parameters.

Boosting, which is a method that can be applied to many statistical methods, expands on this tree-based decision making. With this method, the training data is used to build multiple, sequential trees. After each tree is formed, the next tree essentially learns from the previous iteration by fitting to the residuals of the previous tree. The algorithm, taken from ISLR, is below:

1. Set $\hat{f}(x)= 0$ and $r_i = y_i$ for all i in the training set.

2. For $b = 1, 2, .., B$, repeat:

(a) Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the training data (X, r).

(b) Update $\hat{f}$ by adding in a shrunken version of the new tree: $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$.

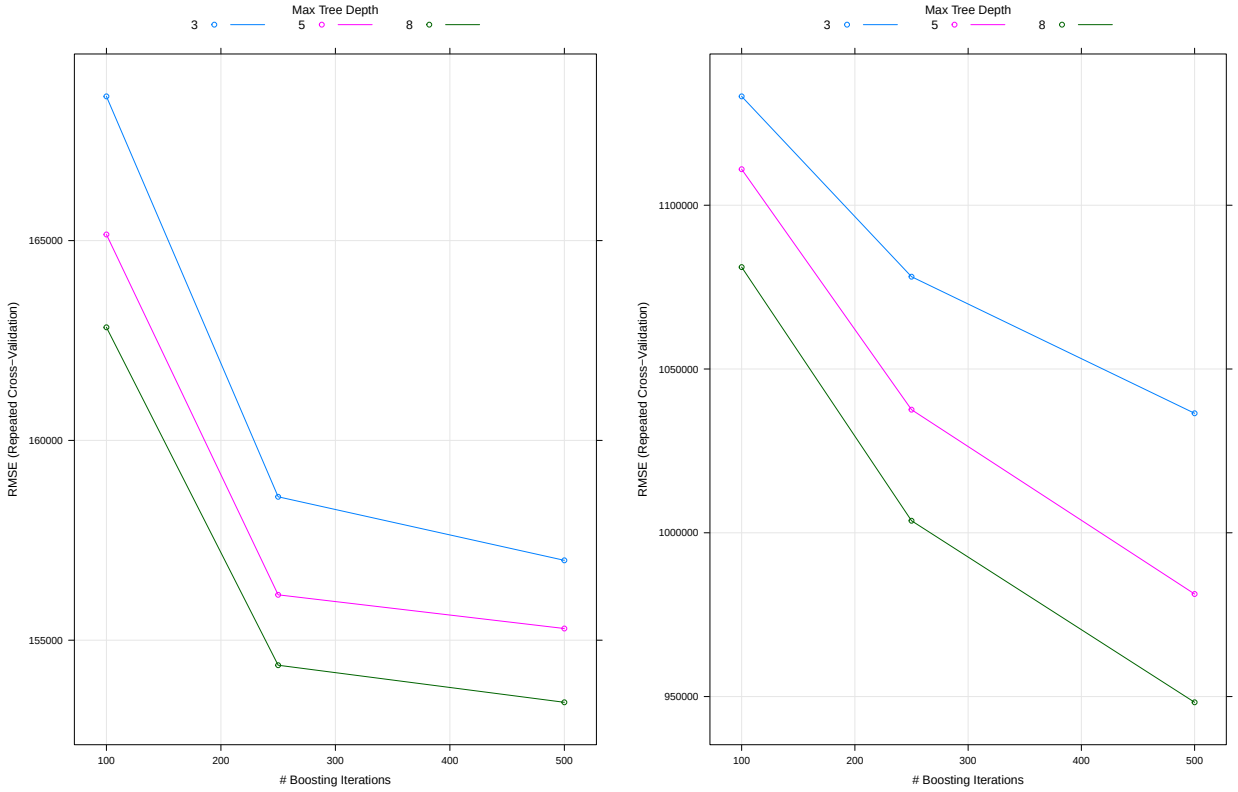
(c) Update the residuals, $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$.

3. Output the boosted model, $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$

I chose this model because it performed best in preliminary versions of all models including random forests, bagging, and a generalized additive model. I ran simple version of each of these models and used them to predict out of sample, choosing the one with the smallest Mean Squared Error. Given more time and computing power, I would have liked to perform cross validation on each of these, but for the purposes of this project I continued with boosting. When boosting, I used interaction terms for all of the independent variables as well.

Tuning Parameters

There are four tuning parameters available in the "gbm" method of training. First is interaction depth. This describes the maximum depth of each tree or the highest variable interaction for the function. The second is the number of trees, which can lead to overfitting but rarely does. The third is shrinkage, which is the λ or learning rate in the above equation. Fourth is "n.minobsinnode", which is a stopping point for the minimum amount of observations in each terminal node.



Once again, due to computational power and time, I could only perform cross validation on two of these four parameters. It is fairly standard to have a shrinkage rate of .01 and minimum observations per node be set to 10, so each of these were held constant. I varied the interaction depths of 3, 5, and 8, and varied the number of trees of 100, 250, and 500. I did this with 5 folds.

The optimum model for Peru had 500 trees and 8 interaction depth. For Ecuador, the optimum model also had 500 trees and 8 interaction depth.

Building an Interpretive Model – OLS

Ordinary Least Squares (OLS) regression is a standard procedure for fitting linear relationships by finding the least squared residual of a model. It outputs interpretable coefficients to understand the variation in the dependent variable and allows for more interpretability. Unfortunately, due to its oftentimes binding and cumbersome assumptions, OLS has a high degree of bias because it places restrictive limits on complex problems. This will be seen as it has a higher Mean Squared Error compared to boosting below.

The Interpretability/Predictability Trade-Off

While models like neural nets and random forests are often very accurate and provide powerful predictive results, they also function as a black box which is uninterpretable. If a business leader was able to tell that a company would lose much of its profit next year but could not describe why, the model they used would not be very helpful in creating actionable change. On the other hand, with very interpretable models, they often are much less accurate since they rely on simpler assumptions. In this case, boosting performs much better than OLS but only gives "relative importance" measures, whereas OLS gives unit by unit interpretations of each of the independent variables.

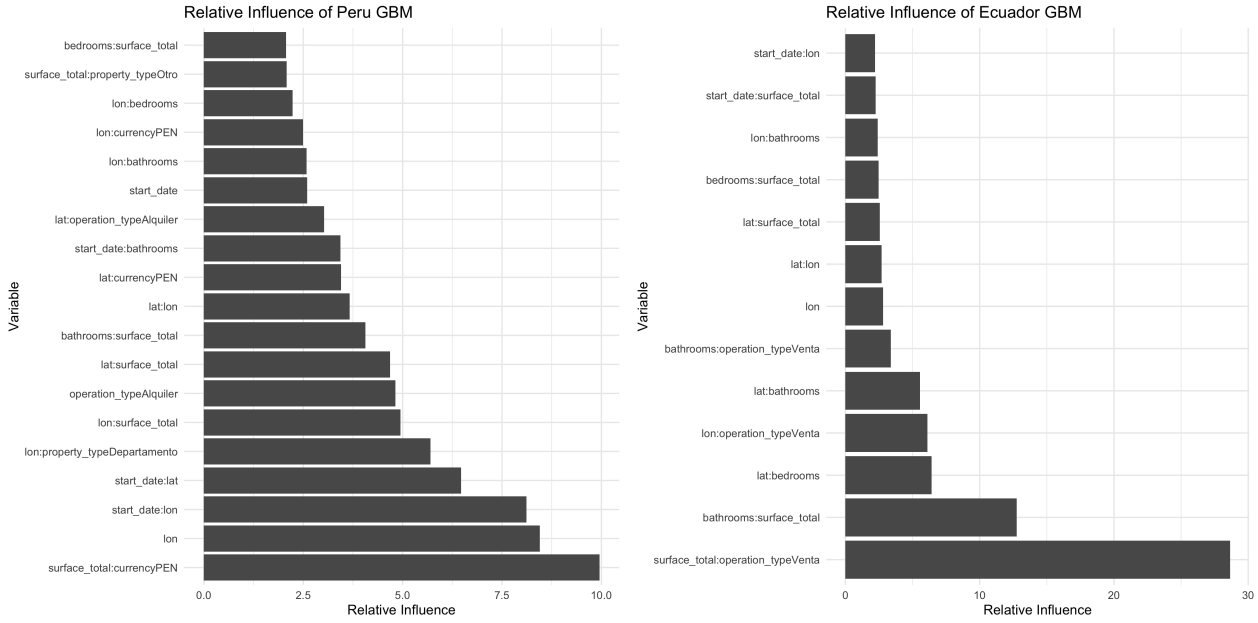
Analysis – Task 2, 3, 4

Drivers of Each Market – Boosting

The boosted regression model was fairly accurate. Both models performed better than their OLS counterparts[2]. The RMSE for Peru's out of sample predictions was 760846.7, which is high, but considering the range of values in the price is understandable. It's R2 score was .61. For Ecuador, the model performed much better out of sample. Its RMSE was 110369.7 with an R2 score of .61.

Boosting, as stated above, only gives some insight into the drivers in each market. It does so by showing the "relative influence" of each independent variable. This measure is created by the "gbm" function. Although helpful at understandign what the model thinks is important, there is little ability to take action on these interpretations since there is no directionality or actual translation to real changes.

[2]See Appendicies



Here, we see that for Peru there are overall more variables of relative influence spread evenly. The first three interactions that are important are surface area total/the currency it was listed in, the longitutde of the property, and the start date/the longitude. This is a good example of why interpretability matters. Although we know that the model uses these interactions of variables in its predictive power, it is difficult to say why the interaction between the day the property was first listed and where the property lies has a large influence on the model. It could be because certain housing markets in a certain time expirienced some change in the price, but it is difficult to know.

For Ecuador, the interaction between the total surface area and the property being sold is by far the most influential. The next is the interaction between total surfact area and the number of bathrooms. The first driver may be explained because there is such a segmented market that surface area may have a larger effect for sales of places over rentals, or the other way around. Similarly for the second driver, the number of bathrooms may have a different effect on price depending on how large the property is.

Drivers of Each Market – OLS

In a more interpretable but less accurate viewpoint, OLS demonstrates different drivers. One important driver for each market represented by different variables is the location of the property. Although not included in the tables below[3], most of the "l2" variables which usually represent the province of the country, are significant in either direction. For Peru, both latitude and longitute are significantly negative, meaning

[3]Full regression results are in Appendix E

the properties that are further south and further east are less expensive. For Ecuador, latitude is negative while longitude is positive meaning properties that are further south and futher west are less expensive.

In Peru, the number of bedrooms is not a statistically significant factor in price. This will be explained later, as I posit that most of this effect is represented in "property type." In Ecuador, the effect is small but negative with a .01% percent decrease. The number of bathrooms is significant in each country, with a 32% increase in price with 1 more bathroom in Peru and a 24% increase in price with one more bathroom in Ecuador. This effect is interesting, but is understandable because this is controlling for all other factors. Bathrooms do not take up much space and can be a valuable addition to a home, so the addition of one may make property price go up signficantly. The surface_total variable, which is the total area in square meters, has a .1% increase in price with every 1,000 square meter increase holding other factors constant.

The type of property does affect the price in differing ways. For example, in Peru, lots ("lote") are 30 percent higher in price than the base level, which is a house ("Casa"). In Ecuador, lots are 78 percent higher on average. The operation type variable, which represents sales of properties ("Venta") and rentals ("Alquiler"), has a relatively large impact compared to property type. On average, sales are 500% more than rentals. Though extreme, this amount makes sense as these property listings for rentals are monthly payments, which are many magnitudes lower than the sale of properties. Since these are such different markets, in the future I would want to run an analysis of each seperately.

Comparing Predictions of Properties

Because the boosted regression model has some worth in predicting housing prices in each of these countries, I compare similar properties and predict their prices in both Peru and Ecuador to find begin to analyze differences in the markets.[4]

First, the mean predicted price in each market varied. For Peru, the average predicted cost was $341,519 while in Ecuador it was $113,281. This was the main difference in many of the predicted values of similar properties, although the difference between predicted prices of similar properties was smaller. For instance, looking at two similar pieces of property[5] in both Ecuador and Peru yielded a difference of around $30,000, with Ecuador's prediction of this property to be around $84,000 and Peru's being $111,000.[6]

[4]Prediction code can be found in Appendix D

[5]Each had 2 bedrooms, 1 bathroom, were in a similar geographic area, had around 85 square meter surface area, was a purchase, was listed less more than 200 days into 2019, and was the "department" property type.

[6]I hoped to simulate my own data and have more control, but unfortunately whenever I created a new dataframe with my own data and used it to predict with the boosted model, R would encounter a terminal error. I attempted to find thses two similar properties by hand instead.

Table 3: Peru OLS Results

	Dependent variable:
	log(price)
start_date	−0.001***
	(0.00004)
lat	−0.361***
	(0.039)
lon	−0.122**
	(0.050)
bedrooms	−0.012***
	(0.003)
bathrooms	0.262***
	(0.004)
surface_total	0.00001***
	(0.00000)
currencyPEN	0.985***
	(0.012)
property_typeLote	0.302***
	(0.085)
property_typeOtro	−0.280***
	(0.015)
property_typeOficina	−0.063*
	(0.037)
property_typeDepartamento	−0.395***
	(0.011)
property_typeLocal comercial	0.210***
	(0.033)
property_typeDepósito	0.797***
	(0.118)
operation_typeAlquiler	−5.181***
	(0.011)
Constant	−1.611
	(4.162)
Observations	27,682
R^2	0.911
Adjusted R^2	0.911
Residual Std. Error	0.684 (df = 27644)
F Statistic	7,623.318*** (df = 37; 27644)

Note: "l2" factors not included 8 *p<0.1; **p<0.05; ***p<0.01

Table 4: Ecuador OLS Results

	Dependent variable:
	log(price)
start_date	−0.00000
	(0.00004)
lat	−0.412***
	(0.039)
lon	0.345***
	(0.018)
bedrooms	0.002
	(0.003)
bathrooms	0.249***
	(0.003)
surface_total	0.00004***
	(0.00000)
property_typeDepartamento	−0.076
	(0.105)
property_typeCasa	−0.147
	(0.105)
property_typeLote	0.738***
	(0.242)
property_typeOtro	−0.118
	(0.105)
property_typeLocal comercial	0.337**
	(0.144)
property_typeDepósito	1.358***
	(0.348)
property_typeCasa de campo	0.733
	(0.585)
operation_typeVenta	5.248***
	(0.006)
Constant	32.999***
	(1.433)
Observations	43,008
R^2	0.954
Adjusted R^2	0.954
Residual Std. Error	0.575 (df = 42971)
F Statistic	24,657.920*** (df = 36; 42971)

Note:"l2" factors not included *p<0.1; **p<0.05; ***p<0.01

Discussion – Task 5

Similarities in Markets

Both models can help describe each market. The similarities between Ecuador and Peru that were found in the boosted regression's description of relative influence include location-related variables, such as longitude and latitude, and their interactions with variables like number of bathrooms and surface area. Surface area was also an important factor shared between each of these countries in the boosted regression. Finally, whether or not a property was to be purchased or rented was a signficant factor in determing price.

In the OLS regression, there were several similarities. The magnitude of whether a property was a rental as opposed to a sale was similar and in the same direction for each country. Surface area also had a similar small but positive effect for Peru and Ecuador. Overall, much like the boosted regression, location seemed to be a influential determining factor on price.

Differences in Markets

In the boosted regression, there were several interactions and variables that had different relative importance. The most obvious is the inclusion of the interaction between currency and surface area in Peru. Currency was not included in the Ecuador dataset due to data limitations, but may have been an important factor in determining price. Additionally, in Peru, one of the largest influential interactions was longitude and the department property type, while no such relationship existed in Ecuador.

One main difference in the OLS regression results was that the date a property was posted had a statistically significant result for Peru, but not for Ecuador. There may have been macroeconomic factors affecting Peru that may have made the start date an important factor in determining price. Another suprising difference was the direction of the bedroom variable. For Peru, an increase in the number of bedrooms decreased the price while Ecuador had no effect, but even still the effect for both was small. Additionally, the direction of the longitude varibale was flipped for Peru and Ecuador. The higher longitude in Peru, the lower the price, while the higher the longitude in Ecuador was found to have a positive effect on price. This may take into account regional economic differences.

Conclusion

Housing prices have an effect on many populations and are of interest to many stakeholders. In the case of Peru and Ecuador, I found that Ecuador's housing market overall is at a lower value and holds differing drivers than Peru. Mainly, the date that a property was posted was an influence on price in Peru but not Ecuador, and location matters in differing ways. Even still, location was a signficant factor in both markets and are an effective way of predicting price.

Appendix A - Data Cleaning

```
set.seed(123)


#reads in data as specific data formats
peru_original <- read_csv("pe_properties.csv",
                          col_types = cols(
                            ad_type = col_factor(),
                            currency = col_factor(),
                            l2 = col_factor(),
                            price_period = col_factor(),
                            property_type = col_factor(),
                            operation_type = col_factor()
                          ))

ecu_original <- read_csv("ec_properties.csv",
                         col_types = cols(
                           ad_type = col_factor(),
                           currency = col_factor(),
                           l2 = col_factor(),
                           price_period = col_factor(),
                           property_type = col_factor(),
                           operation_type = col_factor()
                         ))


# Data Cleaning ----------
peru <- peru_original %>%
  #turns start_date into # of days since 01-01-2019
  mutate(start_date = as.numeric(start_date - ymd(20190101))) %>%
  #filters out NA
  filter(!is.na(price),
         !is.na(lat),
```

```
         !is.na(lon),
         !is.na(l2),
         !is.na(bedrooms),
         !is.na(bathrooms),
         !is.na(surface_total),
         !is.na(currency),
         !is.na(property_type),
         !is.na(operation_type),
         price > 0) %>%
  select_if(~ !any(is.na(.))) %>% #removes any columns with NAs
  select(-id,
         -l1,
         -title,
         -description,
         -ad_type,
         -end_date,
         -created_on) #deselects certain columns

ecu <- ecu_original %>%
  mutate(start_date = as.numeric(start_date - ymd(20190101))) %>%
  filter(!is.na(price),
         !is.na(lat),
         !is.na(lon),
         !is.na(l2),
         !is.na(bedrooms),
         !is.na(bathrooms),
         !is.na(surface_total),
         !is.na(property_type),
         !is.na(operation_type),
         price > 0) %>%
  select_if(~ !any(is.na(.))) %>%
  select(-l1,
```

```
         -title,
         -description,
         -ad_type,
         -id,
         -created_on,
         -end_date,
         -currency)

# split in training and testing

trainIndex <- createDataPartition(peru$price, p = .8,
                                  list = FALSE,
                                  times = 1)

pr_train_set <- peru[c(trainIndex),]
pr_test_set <- peru[-trainIndex,]

trainIndex <- createDataPartition(ecu$price, p = .8,
                                  list = FALSE,
                                  times = 1)

ec_train_set <- ecu[c(trainIndex),]
ec_test_set <- ecu[-trainIndex,]
```

Appendix B- Exploration

```r
set.seed(123)


cor(sapply(pr_train_set, as.numeric)) #creates correlation matrix
```

```
##                  start_date           lat           lon           l2     bedrooms
## start_date      1.00000000 -0.0242582486  0.035203975  0.011247172 -0.051042984
## lat            -0.02425825  1.0000000000 -0.894483204 -0.072876121  0.005328511
## lon             0.03520397 -0.8944832044  1.000000000  0.380455252  0.041088936
## l2              0.01124717 -0.0728761211  0.380455252  1.000000000  0.106244075
## bedrooms       -0.05104298  0.0053285107  0.041088936  0.106244075  1.000000000
## bathrooms      -0.03820760  0.0032439906  0.024532145  0.031969151  0.702507659
## surface_total  -0.01122784 -0.0041093755  0.007031277  0.005955220  0.048148011
## price          -0.05475368 -0.0069362697 -0.018710901 -0.060423070  0.148314900
## currency        0.02212086  0.0364585794  0.004316487  0.064700243 -0.114578207
## property_type   0.08915046 -0.0035159827 -0.063287148 -0.125292965 -0.376950707
## operation_type  0.09129417 -0.0002130956  0.008949140  0.006146497 -0.118097939
##                   bathrooms surface_total       price     currency
## start_date     -0.038207599  -0.011227844 -0.05475368  0.022120863
## lat             0.003243991  -0.004109375 -0.00693627  0.036458579
## lon             0.024532145   0.007031277 -0.01871090  0.004316487
## l2              0.031969151   0.005955220 -0.06042307  0.064700243
## bedrooms        0.702507659   0.048148011  0.14831490 -0.114578207
## bathrooms       1.000000000   0.056627343  0.17139593 -0.128700122
## surface_total   0.056627343   1.000000000  0.04141352 -0.011240641
## price           0.171395930   0.041413524  1.00000000  0.101986307
## currency       -0.128700122  -0.011240641  0.10198631  1.000000000
## property_type  -0.333915963  -0.038933637 -0.11360867  0.071933818
## operation_type -0.106532822  -0.011718711 -0.15504918  0.414293338
##                property_type operation_type
## start_date       0.089150459   0.0912941695
## lat             -0.003515983  -0.0002130956
```

```
## lon             -0.063287148   0.0089491403
## l2              -0.125292965   0.0061464970
## bedrooms        -0.376950707  -0.1180979387
## bathrooms       -0.333915963  -0.1065328224
## surface_total   -0.038933637  -0.0117187114
## price           -0.113608669  -0.1550491761
## currency         0.071933818   0.4142933380
## property_type    1.000000000   0.1163933144
## operation_type   0.116393314   1.0000000000
```

```
cor(sapply(ec_train_set, as.numeric)) #creates correlation matrix
```

```
##                   start_date          lat         lon          l2     bedrooms
## start_date      1.0000000000  0.077524262  0.01570092 -0.07659237 -0.008104366
## lat             0.0775242620  1.000000000  0.52655712 -0.54092596 -0.085532547
## lon             0.0157009153  0.526557120  1.00000000 -0.34398269 -0.015754435
## l2             -0.0765923734 -0.540925963 -0.34398269  1.00000000  0.093077952
## bedrooms       -0.0081043663 -0.085532547 -0.01575443  0.09307795  1.000000000
## bathrooms       0.0047435932 -0.070892695 -0.05305815  0.04925455  0.751360876
## surface_total   0.0011796369 -0.013114369  0.01346391  0.02055702  0.155652686
## price           0.0018268285  0.047658614  0.06751990 -0.02459368  0.341785788
## property_type   0.0009088151 -0.101014453 -0.10811824  0.13138192  0.220116962
## operation_type  0.0044197003  0.009776064  0.04214914  0.03105972  0.291559200
##                   bathrooms surface_total        price property_type
## start_date      0.004743593   0.001179637  0.001826828  0.0009088151
## lat            -0.070892695  -0.013114369  0.047658614 -0.1010144535
## lon            -0.053058149   0.013463905  0.067519899 -0.1081182385
## l2              0.049254552   0.020557024 -0.024593683  0.1313819178
## bedrooms        0.751360876   0.155652686  0.341785788  0.2201169620
## bathrooms       1.000000000   0.179328054  0.402669804  0.1744805088
## surface_total   0.179328054   1.000000000  0.155724156  0.0629237267
## price           0.402669804   0.155724156  1.000000000  0.1182213015
## property_type   0.174480509   0.062923727  0.118221301  1.0000000000
```

```
## operation_type  0.227814553   0.057980321  0.427196128  0.1778918665
##                operation_type
## start_date        0.004419700
## lat               0.009776064
## lon               0.042149138
## l2                0.031059718
## bedrooms          0.291559200
## bathrooms         0.227814553
## surface_total     0.057980321
## price             0.427196128
## property_type     0.177891866
## operation_type    1.000000000
```

```
#descriptive statistics
summary(pr_train_set)
```

```
##    start_date         lat               lon                  l2
##  Min.   : 60.0   Min.   :-18.146   Min.   :-81.27   Lima       :21710
##  1st Qu.:129.0   1st Qu.:-12.133   1st Qu.:-77.05   Arequipa   : 2586
##  Median :221.0   Median :-12.097   Median :-77.01   La Libertad:  980
##  Mean   :217.9   Mean   :-12.113   Mean   :-76.62   Piura      :  726
##  3rd Qu.:288.0   3rd Qu.:-12.071   3rd Qu.:-76.96   Callao     :  476
##  Max.   :448.0   Max.   : -3.511   Max.   :-69.20   Lambayeque :  335
##                                                     (Other)    :  869
##     bedrooms       bathrooms      surface_total          price
##  Min.   : 0.00   Min.   : 1.000   Min.   :    10.0   Min.   :      50
##  1st Qu.: 3.00   1st Qu.: 2.000   1st Qu.:    90.0   1st Qu.:   50000
##  Median : 3.00   Median : 3.000   Median :   130.0   Median :  153000
##  Mean   : 3.44   Mean   : 2.845   Mean   :   280.8   Mean   :  342866
##  3rd Qu.: 4.00   3rd Qu.: 3.000   3rd Qu.:   221.0   3rd Qu.:  335000
##  Max.   :45.00   Max.   :20.000   Max.   :320000.0   Max.   :63050000
##
##  currency            property_type             operation_type
```

```
##  USD:22467   Departamento   :16677   Venta            :21173
##  PEN: 5215   Casa           : 6978   Alquiler         : 6509
##  ARS:    0   Otro           : 3040   Alquiler temporal:    0
##              Local comercial:  497
##              Oficina        :  390
##              Lote           :   66
##              (Other)        :   34
```

```
summary(ec_train_set)
```

```
##    start_date         lat              lon                l2
##  Min.   : 62.0   Min.   :-4.343   Min.   :-90.43   Pichincha:17221
##  1st Qu.:136.0   1st Qu.:-2.192   1st Qu.:-79.90   Guayas   :13592
##  Median :205.0   Median :-2.038   Median :-79.01   Azuay    : 7779
##  Mean   :192.7   Mean   :-1.424   Mean   :-79.20   Manabi   : 2250
##  3rd Qu.:236.0   3rd Qu.:-0.189   3rd Qu.:-78.48   Loja     :  443
##  Max.   :311.0   Max.   : 1.061   Max.   :-76.59   El Oro   :  426
##                                                    (Other)  : 1297
##     bedrooms        bathrooms      surface_total          price
##  Min.   : 1.000   Min.   : 1.000   Min.   :    10.0   Min.   :      50
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.:    90.0   1st Qu.:     700
##  Median : 3.000   Median : 3.000   Median :   130.0   Median :   85000
##  Mean   : 3.048   Mean   : 2.816   Mean   :   230.3   Mean   :  112275
##  3rd Qu.: 3.000   3rd Qu.: 3.000   3rd Qu.:   209.2   3rd Qu.:  147098
##  Max.   :30.000   Max.   :20.000   Max.   :110000.0   Max.   :14000000
##
##          property_type             operation_type
##  Casa           :19587   Alquiler         :15376
##  Departamento   :18087   Venta            :27632
##  Otro           : 5259   Alquiler temporal:    0
##  Local comercial:   34
##  Oficina        :   30
##  Lote           :    7
```

```
##  (Other)        :    4
```

```
# creates tables for report of summary stats
#stargazer(as.data.frame(pr_train_set))
#stargazer(as.data.frame(ec_train_set))
```

Appendix C - Building a Predictive Model (Boosting)

```
set.seed(123)


# cross-validating - TAKES SEVERAL HOURS TO RUN
#-------------------------------------------------
# trainctrl <- trainControl(method = "repeatedcv",
#                           number = 5,
#                           ## repeated ten times
#                           repeats = 1,
#                           verboseIter = TRUE) # selects amount of folds
#
# gbmGrid <-  expand.grid(interaction.depth =c(3,5,8), n.trees = c(100,250,500),
#                         shrinkage =.01, n.minobsinnode = 10) # selects parameters to tune
#
# pr_boost <- train(price~.*.,data=pr_train_set, method = "gbm", distribution= "gaussian",
#                   trControl = trainctrl, tuneGrid = gbmGrid)
#
# ec_boost <- train(price~.*.,data=ec_train_set, method = "gbm", distribution= "gaussian",
#                    trControl = trainctrl, tuneGrid = gbmGrid)



#Runs specific model choice - TAKES AROUND 20 MIN TO RUN
#-------------------------------------------------
#only trains one model
fitControl <- trainControl(method = "none")

#trains model using parameters found through cross-validation
pr_boost <- train(price~.*.,
                  data=pr_train_set,
                  method = "gbm",
                  distribution= "gaussian",
```

```
                  trControl = fitControl,
                  verbose = FALSE,
                  tuneGrid = data.frame(
                    n.trees = 500,
                    interaction.depth = 8,
                    shrinkage = .01,
                    n.minobsinnode = 10))

ec_boost <- train(price~.*.,
                  data=ec_train_set,
                  method = "gbm",
                  distribution= "gaussian",
                  trControl = fitControl,
                  verbose = FALSE,
                  tuneGrid = data.frame(n.trees = 500,
                  shrinkage = .01,
                  interaction.depth = 8,
                  n.minobsinnode = 10))

# saves model
# saveRDS(pr_boost, "./pr_model.rds")
# saveRDS(ec_boost, "./ec_model.rds")

# Creates relative influence graphs
#  ----------------------------
# pr_influence <- summary(pr_boost)
# ec_influence <- summary(ec_boost)

# pr_influence %>%
#   filter(rel.inf > 2) %>% # only get higher influence
#   arrange(rel.inf) %>% # sort by influence
# ggplot(aes(x = rel.inf, y = reorder(var, -rel.inf))) +
```

```
#   geom_bar(stat = "identity") +
#   theme_minimal() +
#   scale_fill_viridis_c() +
#   xlab("Relative Influence") +
#   ylab("Variable") +
#   ggtitle("Relative Influence of Peru GBM") +
#   ggsave("pr_inf.png")
#
# ec_influence %>%
#   filter(rel.inf > 2) %>%
#   arrange(rel.inf) %>%
# ggplot(aes(x = rel.inf, y = reorder(var, -rel.inf))) +
#   geom_bar(stat = "identity") +
#   theme_minimal() +
#   xlab("Relative Influence") +
#   ylab("Variable") +
#   ggtitle("Relative Influence of Ecuador GBM") +
#   ggsave("ec_inf.png")

# loads previously saved models
# pr_boost <- readRDS("pr_model.rds")
# ec_boost <- readRDS("ec_model.rds")

# plots training if cv is ran
#plot(pr_boost)
#plot(ec_boost)
```

```
set.seed(123)
# out of sample testing for boosted regression
pr_boost_pred_is <- predict(pr_boost, pr_train_set) #in sampele
pr_boost_pred_oos <- predict(pr_boost, pr_test_set) # out of sample
RMSE(pr_boost_pred_oos, pr_test_set$price)
```

```
## [1] 760846.7
```

```
R2(pr_boost_pred_oos, pr_test_set$price)
```

```
## [1] 0.615748
```

```
ec_boost_pred_is <- predict(ec_boost, ec_train_set)
ec_boost_pred_oos <- predict(ec_boost, ec_test_set)
RMSE(ec_boost_pred_oos, ec_test_set$price)
```

```
## [1] 142017.6
```

```
R2(ec_boost_pred_oos, ec_test_set$price)
```

```
## [1] 0.6118258
```

Appendix D - Predictions

```r
set.seed(123)


#bind predictions of training price back to training set
pr_predictions <- cbind(pr_train_set, pr_boost_pred_is)
ec_predictions <- cbind(ec_train_set, ec_boost_pred_is)


#find average of the predictions (out of sample)
mean(pr_boost_pred_oos)
```

```
## [1] 358971.8
```

```r
mean(ec_boost_pred_oos)
```

```
## [1] 111078.4
```

```r
#find similar properties and their predicted price
ec_predictions %>%
  filter(bedrooms == 2,
         bathrooms == 1,
         lat < -2.9 & lat > -5,
         lon > -82 & lon < -76,
         surface_total > 80 & surface_total < 90,
         operation_type == "Venta",
         property_type == "Departamento",
         start_date > 200) %>%
  select(ec_boost_pred_is)
```

```
##   ec_boost_pred_is
## 1         83889.87
## 2         83889.87
## 3         83889.87
```

```r
pr_predictions %>%
  filter(bedrooms == 2,
         bathrooms == 1,
         lat < -2.9 & lat > -5,
         lon > -82 & lon < -76,
         surface_total > 80 & surface_total < 90,
         operation_type == "Venta",
         property_type == "Departamento",
         start_date >200) %>%
  select(pr_boost_pred_is)
```

```
##   pr_boost_pred_is
## 1         111504.1
```

Appendix E - Building an Interpratable Model (OLS)

```r
set.seed(123)


# runs simple mutlivariate regression
pr_ols <- lm(log(price) ~., pr_train_set)
ec_ols <- lm(log(price) ~., ec_train_set)


#summarizes regression
summary(pr_ols)
```

```
##
## Call:
## lm(formula = log(price) ~ ., data = pr_train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9574 -0.3874 -0.0376  0.3556  5.2766
##
## Coefficients:
##                               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                 -1.611e+00  4.162e+00   -0.387 0.698659
## start_date                  -8.109e-04  4.178e-05  -19.409  < 2e-16 ***
## lat                         -3.611e-01  3.923e-02   -9.204  < 2e-16 ***
## lon                         -1.216e-01  4.974e-02   -2.445 0.014497 *
## l2San Martin                 1.099e+00  2.612e-01    4.207 2.60e-05 ***
## l2Ica                       -1.204e+00  7.888e-02  -15.258  < 2e-16 ***
## l2Puno                      -1.210e+00  2.879e-01   -4.203 2.64e-05 ***
## l2La Libertad                5.194e-01  1.187e-01    4.378 1.20e-05 ***
## l2Piura                      1.383e+00  2.045e-01    6.764 1.37e-11 ***
## l2Tacna                     -1.712e+00  2.630e-01   -6.511 7.59e-11 ***
## l2Arequipa                  -1.347e+00  2.015e-01   -6.685 2.35e-11 ***
## l2Lambayeque                 7.764e-01  1.598e-01    4.860 1.18e-06 ***
```

```
## l2Moquegua                   -1.951e+00  2.549e-01   -7.656 1.98e-14 ***
## l2Cusco                      -2.145e-01  2.220e-01   -0.966 0.333961
## l2Cajamarca                   6.966e-01  1.649e-01    4.225 2.39e-05 ***
## l2Loreto                      1.903e+00  4.802e-01    3.963 7.43e-05 ***
## l2Tumbes                      1.213e+00  2.917e-01    4.159 3.20e-05 ***
## l2Huánuco                     6.217e-01  1.942e-01    3.201 0.001370 **
## l2Ancash                      1.746e-03  1.466e-01    0.012 0.990498
## l2Callao                     -5.321e-01  3.218e-02  -16.537  < 2e-16 ***
## l2Pasco                       7.433e-01  4.995e-01    1.488 0.136754
## l2Junín                      -6.948e-01  1.479e-01   -4.698 2.64e-06 ***
## l2Ucayali                     8.956e-01  2.694e-01    3.324 0.000887 ***
## l2Amazonas                    3.128e+00  5.250e-01    5.959 2.57e-09 ***
## l2Apurimac                   -1.196e+00  5.125e-01   -2.333 0.019637 *
## l2Madre de Dios              -1.601e+00  5.433e-01   -2.947 0.003211 **
## l2Ayacucho                    3.744e-01  6.938e-01    0.540 0.589445
## bedrooms                     -1.242e-02  2.715e-03   -4.573 4.82e-06 ***
## bathrooms                     2.616e-01  3.591e-03   72.845  < 2e-16 ***
## surface_total                 1.002e-05  1.054e-06    9.499  < 2e-16 ***
## currencyPEN                   9.846e-01  1.173e-02   83.951  < 2e-16 ***
## property_typeLote             3.021e-01  8.527e-02    3.542 0.000397 ***
## property_typeOtro            -2.797e-01  1.519e-02  -18.413  < 2e-16 ***
## property_typeOficina         -6.279e-02  3.706e-02   -1.695 0.090173 .
## property_typeDepartamento    -3.946e-01  1.083e-02  -36.436  < 2e-16 ***
## property_typeLocal comercial  2.096e-01  3.309e-02    6.335 2.41e-10 ***
## property_typeDepósito         7.971e-01  1.183e-01    6.736 1.66e-11 ***
## operation_typeAlquiler       -5.181e+00  1.104e-02 -469.450  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6839 on 27644 degrees of freedom
## Multiple R-squared:  0.9107, Adjusted R-squared:  0.9106
## F-statistic:  7623 on 37 and 27644 DF,  p-value: < 2.2e-16
```

```r
summary(ec_ols)
```

```
##
## Call:
## lm(formula = log(price) ~ ., data = ec_train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3177 -0.3299 -0.0543  0.2624  5.4801
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       3.300e+01  1.433e+00  23.023  < 2e-16 ***
## start_date                       -3.833e-06  4.075e-05  -0.094 0.925055
## lat                              -4.119e-01  3.945e-02 -10.439  < 2e-16 ***
## lon                               3.448e-01  1.825e-02  18.891  < 2e-16 ***
## l2Loja                           -1.674e+00  1.500e-01 -11.155  < 2e-16 ***
## l2Guayas                         -4.764e-01  7.380e-02  -6.455 1.09e-10 ***
## l2Esmeraldas                      7.370e-01  6.717e-02  10.973  < 2e-16 ***
## l2Chimborazo                     -8.933e-01  6.743e-02 -13.248  < 2e-16 ***
## l2Azuay                          -1.092e+00  1.049e-01 -10.412  < 2e-16 ***
## l2Manabi                          2.008e-01  4.389e-02   4.575 4.77e-06 ***
## l2Carchi                          1.785e+00  2.589e-01   6.898 5.36e-12 ***
## l2Cañar                          -9.190e-01  1.056e-01  -8.704  < 2e-16 ***
## l2El Oro                         -1.135e+00  1.199e-01  -9.465  < 2e-16 ***
## l2Santo Domingo De Los Tsáchilas -2.471e-01  4.069e-01  -0.607 0.543768
## l2Cotopaxi                       -3.689e-01  7.479e-02  -4.933 8.14e-07 ***
## l2Tungurahua                     -6.765e-01  7.518e-02  -8.998  < 2e-16 ***
## l2Imbabura                       -1.148e-01  3.604e-02  -3.184 0.001453 **
## l2Pastaza                        -9.791e-01  2.929e-01  -3.343 0.000831 ***
## l2Los Rios                       -9.332e-01  1.503e-01  -6.210 5.34e-10 ***
## l2Sucumbios                      -1.351e+00  4.076e-01  -3.316 0.000915 ***
## l2Morona Santiago                -1.190e+00  1.927e-01  -6.174 6.74e-10 ***
```

```
## l2Galapagos                          4.303e+00  3.329e-01  12.924  < 2e-16 ***
## l2Bolivar                           -1.423e+00  4.103e-01  -3.468 0.000526 ***
## l2Zamora Chinchipe                  -1.665e+00  2.776e-01  -5.998 2.01e-09 ***
## l2Orellana                          -1.474e+00  1.572e-01  -9.379  < 2e-16 ***
## l2Napo                               6.892e-01  2.894e-01   2.382 0.017242 *
## bedrooms                             2.069e-03  2.897e-03   0.714 0.475188
## bathrooms                            2.493e-01  2.921e-03  85.350  < 2e-16 ***
## surface_total                        3.835e-05  2.700e-06  14.204  < 2e-16 ***
## property_typeDepartamento           -7.614e-02  1.052e-01  -0.724 0.469117
## property_typeCasa                   -1.469e-01  1.052e-01  -1.397 0.162486
## property_typeLote                    7.381e-01  2.415e-01   3.056 0.002245 **
## property_typeOtro                   -1.178e-01  1.054e-01  -1.117 0.263852
## property_typeLocal comercial         3.366e-01  1.445e-01   2.330 0.019811 *
## property_typeDepósito                1.358e+00  3.484e-01   3.897 9.76e-05 ***
## property_typeCasa de campo           7.327e-01  5.850e-01   1.252 0.210395
## operation_typeVenta                  5.248e+00  6.365e-03 824.599  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5752 on 42971 degrees of freedom
## Multiple R-squared:  0.9538, Adjusted R-squared:  0.9538
## F-statistic: 2.466e+04 on 36 and 42971 DF,  p-value: < 2.2e-16
```

```r
# creates coefficient table
#-----------------------
# stargazer(pr_ols)
# stargazer(ec_ols)


#predict out of sample
pr_ols_pred_oos <- predict(pr_ols, pr_test_set)
pr_ols_pred_oos <- exp(pr_ols_pred_oos) #convert back to actual units, was in log
RMSE(pr_ols_pred_oos, pr_test_set$price) #calc RMSE
```

```
## [1] 1138252
```

```
R2(pr_ols_pred_oos, pr_test_set$price) # calc R2
```

```
## [1] 0.09507032
```

```
ec_ols_pred_oos <- predict(ec_ols, ec_test_set)
ec_ols_pred_oos <- exp(ec_ols_pred_oos)
RMSE(ec_ols_pred_oos, ec_test_set$price)
```

```
## [1] 191521.5
```

```
R2(ec_ols_pred_oos, ec_test_set$price)
```

```
## [1] 0.2440026
```