Exploring Peruvian and Ecuadorian housing prices

Jingyi Cui

2020/06/06

Contents

1 Introduction

Housing prices are affected by a lot of factors, such as government policies, GDP, etc. For example, during this period, people are experiencing a hard time with COVID-19. The housing prices may fall down, since less people may have plans for investing. In this report, we do not analyze those external factors; instead, we analyze how internal factors like number of bedrooms or bathrooms affect the prices. We will finally predict the housing prices using different methods.

We will focus on two datasets: Peruvian and Ecuadorian housing prices, including 25 variables. We will analyze the features, compare the two datasets with some predicative models and predict the price.

2 Data Description

Both datasets have same variables. Below is the description taken from the original Kaggle site given above:

- id - id of the listing. It is not unique: if the listing is updated by the real estate agency (new version of the listing) a new record is created with the same id but different dates: registration and cancellation.
- start_date - Date of registration of the listing.
- end_date - Cancellation date of the listing.
- created_on - Date of registration of the first version of the listing.
- lat - Latitude of the property.
- lon - Longitude of the property.
- l1 - Administrative Level 1: Country of the property.
- l2 - Administrative Level 2: Usually the province of the property.
- l3 - Administrative Level 3: Usually the city of the property.
- l4 - Administrative Level 4: Usually the neighbourhood of the property.
- operation - Type of listing:
  - Venta (Sale).
  - Alquiler (Rent).
- type - Type of property:
  - Casa (House).
  - Departamento (Apartment).
  - PH (Horizontal Property).
- rooms - Number of rooms (useful for Argentina, which is not our target dataset).
- bedrooms - Number of bedrooms (useful for the rest of the countries).
- bathrooms - Number of bathrooms.
- surface_total - Total area in m^2.
- surface_covered - Area covered in m^2.
- price - Price published in the listing.
- currency - Currency of published price.

- price_period - Payment periods:
  - Diario (Daily).
  - Semanal (Weekly).
  - Mensual (Monthly).
- title - Title of the listing (These are in Spanish).
- description - Description of the listing (In Spanish).
- status - Development status (Completed, Under construction, . . .).
- name - Development name.
- short_description - Short listing description.

2.1 Feature Observation

According to our assumption and initial observation, we make the following assumptions:

- Houses with more total are and bedrooms (more "surface_total" and "bedrooms") will worth more. Usually, much bigger room can contain more people, so it is appropriate and reasonable that they are worthy. They are in positive relationship.
- Houses with type being "House" will be worthy than Apartment and Horizontal Property. It is likely that in neighborhoods with houses, the views may be good and it is less crowded. People tend to have their own places if they earn more money. Since the type is a categorical variable, there does not exist some positive or negative relationship.

We will find out if these assumption are correct for each dataset through the project.

For better analyzing the dataset, we notice some of the variables are useless, such as title, id, etc. In this report, I only use lat(latitude), lon(longitude), bedrooms, bathrooms, surface_total, surface_covered to analyze the price. Following our basic assumptions, we believe those six variables are useful and helpful to predict the price.

Another thing that is needed to be pointed out is these two datasets are not very clean. There exist tons of NA in the original dataset, which makes the analysis be really hard. I replace all the NA values with the mean of that column for processing the analysis.

2.2 Peruvian

This dataset contains 124449 observations and 25 variables. We first take a look at the log distribution of the housing prices(Figure 1), since it looks better than version without log: it has two peaks. Log prices make each peak looks like normal better for visualisation. In addition, we have the correlation plot (Figure 2) between bedrooms, bathrooms, surface_total, surface_covered and price (due to high volume of data, I only use the first 5000 observations). We do not observe a clear relationship between those variables and price; however, we will do some analysis using all the data later.

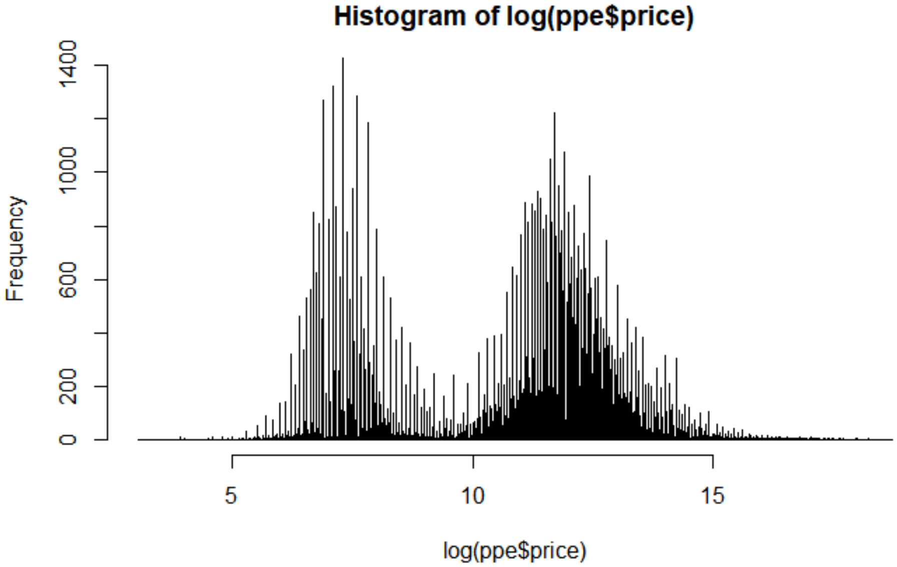

Figure 1: Histogram of Peruvian log housing prices

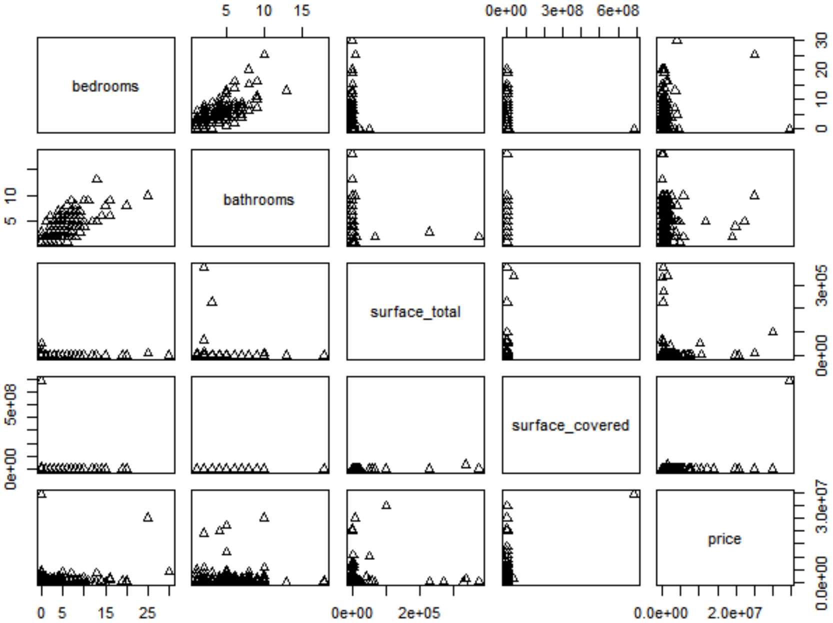

Figure 2: Correlation plot

2.3 Ecuadorian

This data sets have 143565 observations and 25 variables. We first see how the housing prices are distributed (Figure 3). We see that the log prices also have two peaks. By the summary of the housing prices, we know that minimum of housing price is 0 and maximum is 40,000,000. There are 1609 missing values for prices. Next, let us look at the correlation plot between bedrooms, bathrooms, surface_total, surface_covered and price (I only use the first 5000 observations for reduction of running time). The relationship is not clear though, and we will explore it more in later analysis.

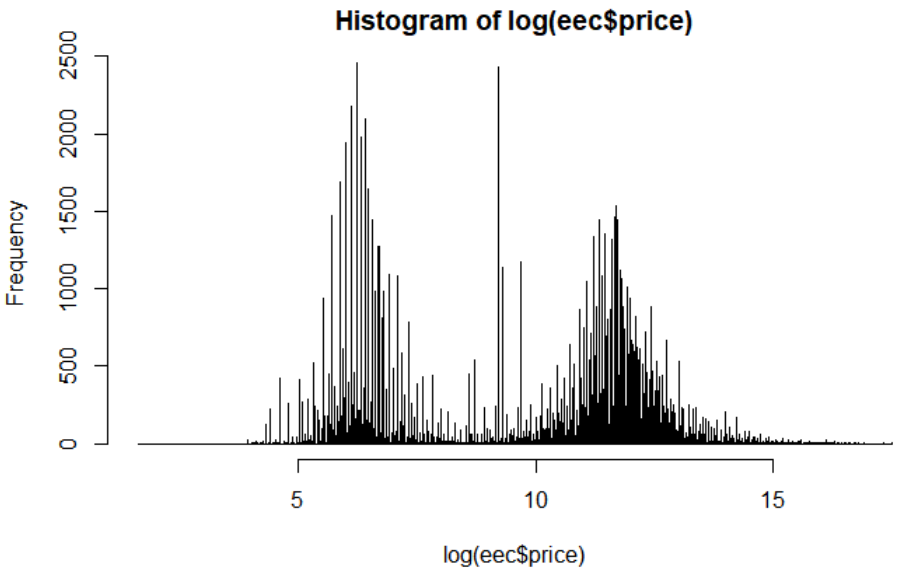

Figure 3: Histogram of Ecuadorian log housing prices

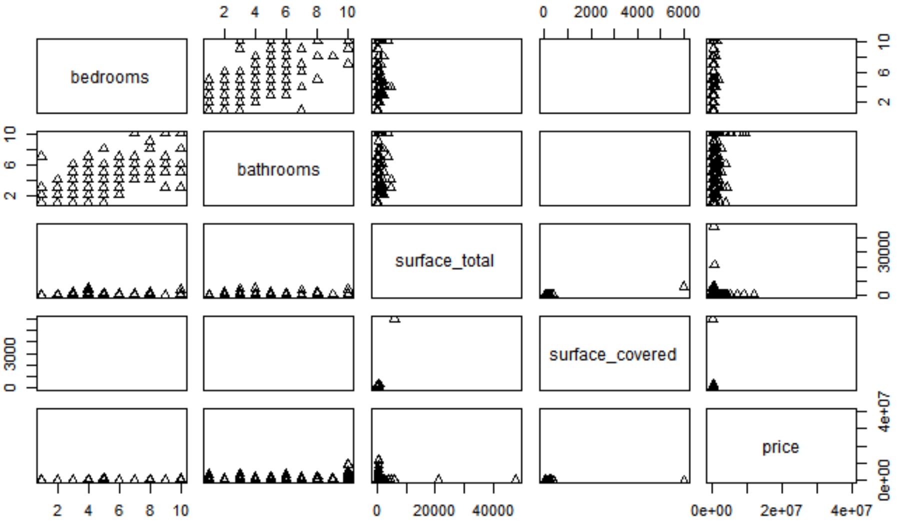

Figure 4: Correlation Plot

3 Methodology and Conclusion

As we know, there exist trade-off between interpretability and accuracy, shown in the picture below. As accuracy increases, we can interpret the model and the result less clearly; for example, there exist a lot hidden layers in neural network, so it is hard for us to interpret how it works or relationship between two variables. For predicative model like decision tree, people can easily read the tree, like how the tree starts and ends, number of nodes in the tree. For linear regression, people can easily find the coefficient in front of each variable and observe if the prices increase or decrease directly.

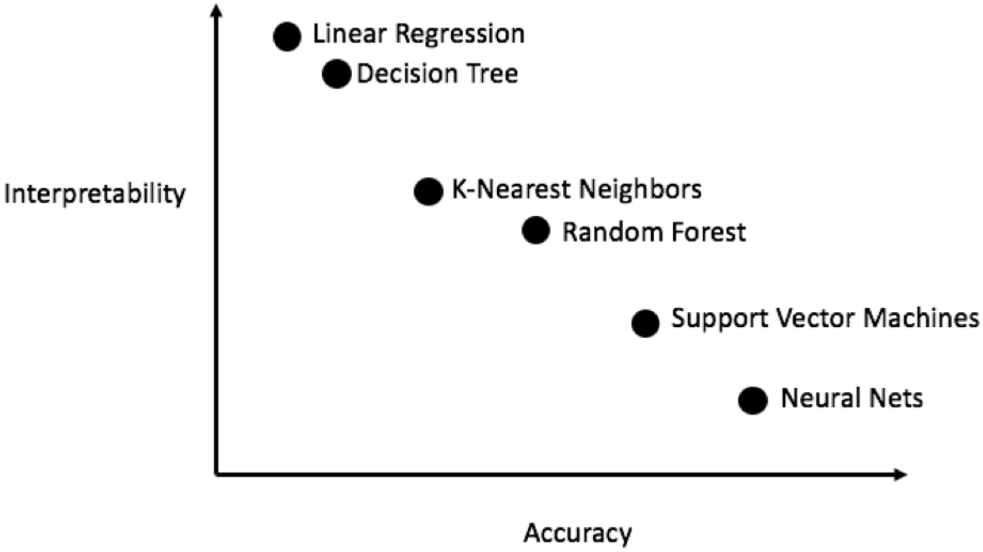

Figure 5: Tradeoff between interpretability an accuracy

Since this is a regression problem, our choices are limited: linear regression, ridge regression, lasso regression, decision trees, random forest, adaboost and K-Nearest Neighborhood. Also, we try fit some interpretable models. We will focus methods above except adaboost.

3.1 Peruvian

I try six methods to analyze Peruvian dataset and then calculated root mean square error (RMSE). I found the result corresponds to the "Interpretability vs Accuracy". The RMSE for linear model, ridge regression, lasso regression, decision tree and KNN are 1774550, 1774548. 1774547, 1757830 and 1827131.

Surprisingly, the RMSE of KNN is the largest. One reason I guess is because the k I choose is not the best. Smaller k may not give us a good answer; at the same time, a larger k may result in overfitting. More importantly, We know that KNN algorithm is to use the most k nearest neighbor and classify them to get the type of our target. It is easily by the choice of k and the data itself. Since I replace all the missing values with the mean of the column that the missing value belongs to. Thus, the housing prices will be easily by neighbors, which may be original NA.

Among these six methods, the best is decision tree. Even though I did not select best tuning parameter, such as number of nodes to choose when to stop, it still behaves good. We need to be awared of overfitting problem, since a lot of nodes may result a good accuracy. However, if we apply this model to other datasets, it may generate terrible answer. Choosing best tuning parameters is what we need to do later.

Linear model, ridge regression and lasso regression are almost the same, though existing some tuning parameter. Using different lambda for ridge regression result in different RMSE. In this case, as lambda increases, RMSE increases. However, we need to notice that if we increase lambda too much, there may generate overfitting problem.

3.2 Ecuadorian

Similarly, I use five different model to fit this dataset and make predictions. Then, root mean squared error is calculated. RMSE, for linear model, ridge regression, lasso regression and decision tree are 572780, 572780.5, 572780.1, 573298.2. These values have no such big difference .

We can find that the difference among linear regression, ridge regression and lasso regression are really small. This may because the parameter I choose, such as various alpha for ridge and lasso regression. Surprisingly, the RMSE of decision tree is the largest. This may have different reasons, maybe because the missing values are replaced by mean, having an effect on the whole dataset, fitting and prediction. Or maybe because I did not choose the best tuning parameters: the tree stops earlier and too later, resulting in underfitting and overfitting problem.

Seeing the results of these two datasets, we find that there exist some similarities with these two datasets. When we use the models whose interpretability is high, RMSE values are really similar. This may because the datasets have a lot of features in common. Importantly, from the histogram of log housing prices, we can see that both datasets have two peaks. This implies that if we use similar model to predict the future housing price, the prices may have similar shapes (I will draw the histogram of predicative prices). Also, the way that I deal with missing values are the same. I am not sure if these will bring similar results to different models. All these illustrate that the two markets are similar.

4 Discussion

Even though the result I get corresponds to my assumption, there still exist some points that can be improved. The first problem is how to handle NA values. Fitting some models does not need to deal with missing values. However, the algorithm like K-Nearest Neighbor cannot make full use of missing values, since it cannot classify the target to be NA. In this project, I treat all the missing values to be the mean of the column that the missing value belongs to. Thus, there exist a lot of same values in each column, which may affect our result. for example, for ridge and lasso, while normalizing, the missing values will be normalized to be 0. I think the best way is to omit the columns hat have missing values. However, in these two datasets, there are too many missing values.

The second problem is the high volume of the datasets. There two datasets have more than 120,000 and 140,000 rows respectively. The running time are too long to be estimated. Importing the dataset may spend more than 1 minute, not to mention training the dataset using different models. For example, doing random forest may need more than 2 minutes. The worst result is the program runs so long that cannot even get the answer. I did not set the validation dataset. I think it may spend longer if I choose to get the best tuning parameter for models like decision trees.

Thirdly, I did not select any categorical features for doing analysis. But I guess the variables like status may affect the housing price. In later modelling, I may try to add more features to continue to analysis. Due to the limitation of the language the dataset uses, I find it hard to classify and understand the dataset. This problem can be solved later. Lastly, there exist different currencies for the housing prices and it should be transofrmed to the same currency.

5 Appendix

5.1 Peruvian

~~~~
# Import Dataset
```{r}
dir <- "C:/Users/cuijy/Desktop"
pe <- read.csv(file.path(dir, "pe.csv"))
```

# Choose a few useful features to analyze
```{r}
ppe <- pe[, c("lat", "lon",  "bedrooms", "bathrooms",
~~~~

```
                "surface_total", "surface_covered", "price")]
‘‘‘

# Summary Statistic
‘‘‘{r}
#names(ppe)
summary(ppe$price)
hist(log(ppe$price),breaks = 100000)
pairs(ppe[1:5000,3:7], pch = 2)
‘‘‘

# Replace missing value with mean
‘‘‘{r}
for(i in 1:ncol(ppe)){
  ppe[is.na(ppe[,i]), i] <- mean(ppe[,i], na.rm = TRUE)
}
‘‘‘

# Split dataset
‘‘‘{r}
set.seed(123)
## will train the model only on 80% of the whole sample
rand =runif(length(ppe$price), min = 0, max = 1)
train=rep(FALSE,length(ppe$price))
train[rand<=0.8]=TRUE
test = !train

train_data = ppe[train,]
test_data = ppe[!train,]
‘‘‘

‘‘‘{r}
library(Metrics)
‘‘‘

# PE
## Linear Model
‘‘‘{r}
lm_pe <- lm(price ~ ., data = train_data)
# summary(lm_pe)
lm.pr <- predict(lm_pe, test_data)
# summary(lm.pr)
rmse(lm.pr, test_data$price)
‘‘‘

## Ridge Regression
‘‘‘{r}
x = ppe[, 1:6]
grid = 10^seq(-2, 10, length = 12)
ridge = glmnet(as.matrix(x[train,]), as.matrix(train_data$price), alpha = 0,
                 lambda = grid, thresh = 1e-12)
ridge.pr = predict(ridge, s = 1000, newx = as.matrix(x[test,]))
rmse(ridge.pr, test_data$price)
```

~~~
```

## Lasso Regression
```{r}
lasso = glmnet(as.matrix(x[train,]), as.matrix(train_data$price), alpha = 1,
            lambda = grid)
lasso.pr = predict(lasso, s = 4, newx = as.matrix(x[test,]))
rmse(lasso.pr, test_data$price)
#plot(lasso)
```

## Decision Tree
```{r}
library(tree)
```
```{r}
tre = tree(price~., data = train_data)
tre.pr = predict(tre, test_data)
rmse(tre.pr, test_data$price)
```

## KNN
```{r}
library(class)
```
```{r}
k = round(sqrt(nrow(ppe)))
knn.pr = knn(train_data, test_data, train_data$price, k)
rmse(as.numeric(knn.pr), test_data$price)
```
~~~

5.2 Ecuadorian

~~~
# Import Dataset
```{r}
dir <- "C:/Users/cuijy/Desktop"
ec <- read.csv(file.path(dir, "ec.csv"))
```

# Choose a few useful features to analyze
```{r}
eec <- ec[, c("lat", "lon",  "bedrooms", "bathrooms",
                    "surface_total", "surface_covered", "price")]
```

# Summary Statistic
```{r}
#names(eec)
summary(eec$price)
hist(log(eec$price), breaks = 100000)
pairs(eec[1:5000,3:7], pch = 2)
```

# Replace missing value with mean
~~~

~~~
```{r}
for(i in 1:ncol(eec)){
  eec[is.na(eec[,i]), i] <- mean(eec[,i], na.rm = TRUE)
}
```

# Split dataset
```{r}
set.seed(123)
## will train the model only on 80% of the whole sample
rand = runif(length(eec$price), min = 0, max = 1)
train = rep(FALSE, length(eec$price))
train[rand <= 0.8] = TRUE
test = !train

train_data = eec[train,]
test_data = eec[!train,]
```

```{r}
library(Metrics)
```



# EC
## Linear Model
```{r}
lm_ec <- lm(price ~ ., data = train_data)
# summary(lm_ec)
lm.pr <- predict(lm_ec, test_data)
# summary(lm.pr)
rmse(lm.pr, test_data$price)
```

## Ridge Regression
```{r}
x = eec[, 1:6]
grid = 10^seq(-2, 10, length = 12)
ridge = glmnet(as.matrix(x[train,]), as.matrix(train_data$price),
               alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pr = predict(ridge, s = 1000, newx = as.matrix(x[test,]))
rmse(ridge.pr, test_data$price)
```

## Lasso Regression
```{r}
lasso = glmnet(as.matrix(x[train,]), as.matrix(train_data$price),
             alpha = 1, lambda = grid)
lasso.pr = predict(lasso, s = 4, newx = as.matrix(x[test,]))
rmse(lasso.pr, test_data$price)
#plot(lasso)
```
~~~

```
## Decision Tree
‘‘‘{r}
library(tree)
‘‘‘
‘‘‘{r}
tre = tree(price~., data = train_data)
tre.pr = predict(tre, test_data)
rmse(tre.pr, test_data$price)
```