Battle of Neighborhoods in Barcelona

Yueming HU
April 2, 2021

1 Introduction

1.1 Background

Barcelona is a very nice city located in the coastal area of northeastern Spain.

Client L is a newly established real estate company based in Barcelona. This company intends to develop several real estate projects in different neighborhoods of the city. As we know, property project needs to consider its surrounding environment, what facilities and venues are nearby. And the prices of real estates can differ greatly according to their location, size, room numbers, etc.

1.2 Problem

Client L is not familiar with all the neighborhoods in Barcelona and also not sure what should be the reasonable prices to set for properties built in future in different neighborhoods. Therefore, the company entrusts me to provide a solution by using data analysis, so as to allow them to choose appropriate neighborhoods in a better-informed manner and have a better ideal of the possible price.

After initial analyzing of Client L's request, I decide to divide this problem into two parts, getting a general overview of neighborhoods' characteristics; and predicting the housing price based on different features, such as location (neighborhood), housing size, room numbers, floor numbers, etc.

1.3 Interest

Obviously, real estate companies would be interested in the prediction of housing prices and neighborhoods characteristics. The solution can even be used for property development in different cities in future.

Potential housing buyers may also be interested in, because this solution can help them choosing neighborhoods to live in and checking whether the price is reasonable or not.

2 Data Acquisition and Cleaning

2.1 Data sources

The housing price data of Barcelona can be found in a Kaggle dataset, at https://www.kaggle.com/jorgeglez/barcelona-idealista-housingprices .

Neighborhood information is got from Foursquare Api.

To use Foursquare Api request, geographical coordinates, i.e. latitude and longitude of all the neighborhood are necessary, so the latitude and longitude are obtained through Geopy.

2.2 Data cleaning

The Kaggle's csv file is read into dataframe by pandas. There are 3265 samples and 12 columns. Among them, the "floor" column is recorded as string, such as "floor 1" and "ground floor". I need to change this column into integer so as to do analysis.

I check all the unique values of “floor column”, and find that there are “mezzanine”, “ground floor” and “Multiple floor”. I replace the “ground floor” with “floor 0”, “mezzanine” with “floor 0.5”, and “Mutiple floor” with “floor 1.5”. Because mezzanine is the one between two floors and mutiple floor may refer to the duplex, which usually has two or three stories, here I just use the floats to show them as different types.

Then I use lambda function to split the word “floor” and the number, keep only the number (integer) into a separate column as “floor number”.

2.3 Feature Selection

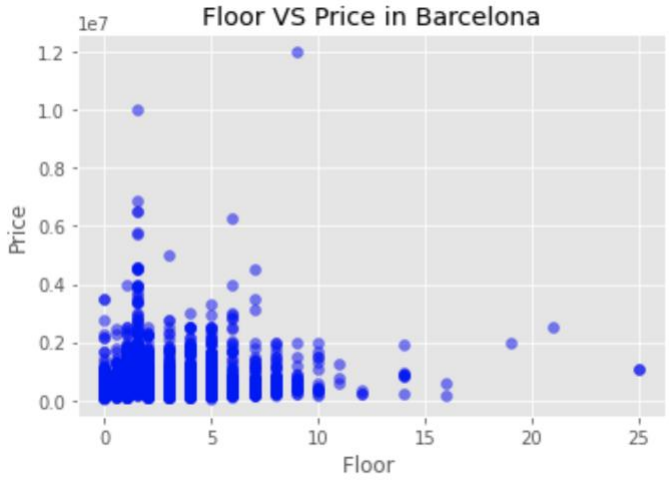
The features in the dataframe are: “city”, “district”, “neighborhood”, “condition”, “type”, “rooms”, “area_m2”, “lift”, “views”, “floor number”, “prices”. Since my prediction target is a reasonable price of property, uased as reference for my client, so I will choose only the very typical features, which means those have relatively big or obvious impact on the price.

“Neighborhood”, no need to say, refers to the location, is a decisive factor to price, “area”, “room number” are also important features. In contrast, housing type, whether there is a lift or not, and with or without exterior view are not that decisive, also they are string type, so I will not use them. But I’m not sure about the features of condition and floor number. So I examine them in more detail.

For “condition”, the unique values are “good condition”, “new”, “needs refurb”. In order to see whether different conditions lead to huge difference in price, I decide to plot them onto an image to have a direct view. I use blue dots to represent “good condition”, yellow dots as “needs refurb”, green as “new”. The plot is as below.

From the plot, we can see that the dots of different colors are mixed together, especially new and good condition, and sometimes for the same area, the needs refurb are even more expensive than the good condition. This means condition is not a decisive factor to the price, and it will not cause huge difference in prices.

For floor number, the usual understanding is that the higher the floor is, the more expensive the price, because the view would be better. I also use image to have a look. The x-axis is floor number, and y-axis is price. The plot is as below.



From the plot, out of my expectation, there is no strong relationship between the floor number and price. For each different floor below 10th floors, there can be very high and very low price. The line of floor 1.5 has prices higher than others, but it refers to duplex, so the main reason behind may be the size. Except for this floor, the price range of others are mostly the same. So, the floor feature will also not be used in this model building for a general prediction.

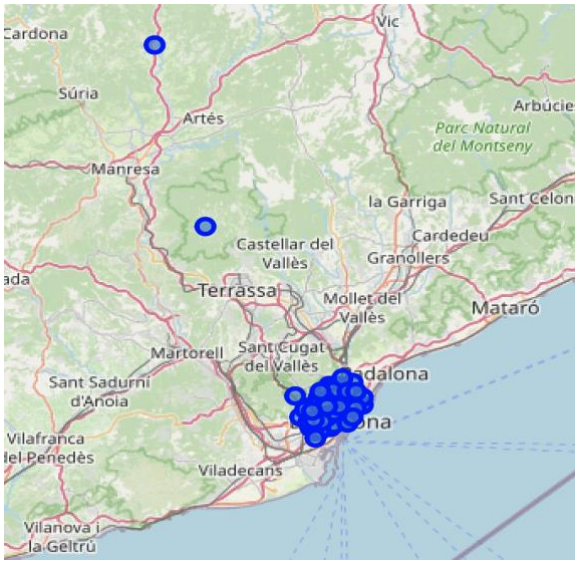
As for the neighborhoods data, there altogether 65 neighborhoods. But when I use grouby and count method to check the data, I find that several neighborhoods have only three or four samples. I think if a neighborhood has very little sample data, the prices for this neighborhood would not be meaningful. So I called out all the neighborhoods with 10 or less than 10 samples, using a lambda function to mark them as category “other”. There are all together 56 samples of “other”, and I drop them.

Now, there left 3207 samples of 55 neighborhoods. Since neighborhoods feature is string type, I use one-hot encoding to convert all the neighborhoods into dummy variables.

2.4 Removing and correcting outliers

From the above plots, I noticed there is dot showing extremely high price, which is in fact the highest price. So I take a look and find that the price of the sample is 12000000, area is 600 m^2 and only four rooms. It’s not possible for a 600 m^2 house having only 4 rooms and the price is really ridiculous. I think this sample is not correct and remove it.

For latitude and longitude data, I use Geopy’s Nominatim function to get the geocodes. Since this way uses the name of location to get the geocodes, there might be inaccurate data. So I plot all the neighborhoods on the map to see if all the location data is right. The map is as below.



Barcelona is near the coast, but two dots are far away from the coastal area. I modify the address of the two neighborhoods respectively, make request again, and then get all correct data. The correct map is as below.