Real estate pricing in Paris

A data-oriented price estimation approach

Augustin CURLIER

Introduction/Business Problem

In a heterogeneous and pricy city such as Paris, France, it might be complicated for buyers to estimate if the price of the apartment they wish to acquire is fair, or for seller to estimate the correct price to which advertise their good.

Price per square meter provided by real-estate professionals is often an averaged value over large administrative districts 'arrondissement'. These estimations do not reflect the fine particularities of neighborhoods, which might have a large impart on the price.

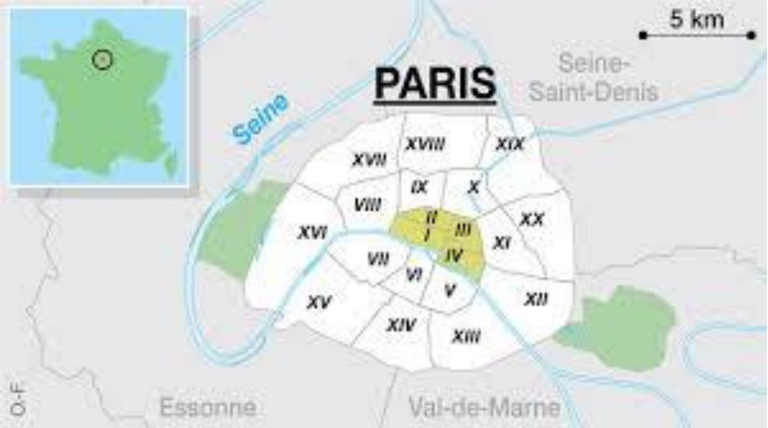


Figure 1: Paris districts (20 in total)

The idea here is to develop a model based on publicly available data and machine learning algorithms to come up with a predictor able to yield an estimation of the price per square meter for a Parisian apartment, taking into account more element than just the district onto which the apartment is located.

Data

In order to setup this estimator, the model will rely on a few publicly available data sources.

First is the list of all transactions completed in France for the year 2018. This dataset is publicly available and counts a few millions real estate transactions. This will be the primary source of information used to build the model :https://www.data.gouv.fr/fr/datasets/r/1be77ca5-dc1b-4e50-af2b-0240147e0346

The second data source used is the official sub-division of paris into smaller administrative districts (Quartiers). This data is used in the form of a geoJSON file and will allow us to present the results using a more interesting meshing than the coarse administrative districts maps : https://opendata.paris.fr/explore/dataset/quartier_paris/download/?format=geojson&timezone=Europe/Berlin

The third data source is OpenStreetMaps, through its tool to search OSM data by name and address (geocoding), called Nominatim. It can be found at nominatim.openstreetmap.org. The GeoPy library, which is a Python 2 and 3 client for several popular geocoding web services allows to access Nominatim though just a few lines of code. This data will allow to acquire the position of the apartments which have been sold in 2018 using their postal addresses as provided in the transaction dataset.

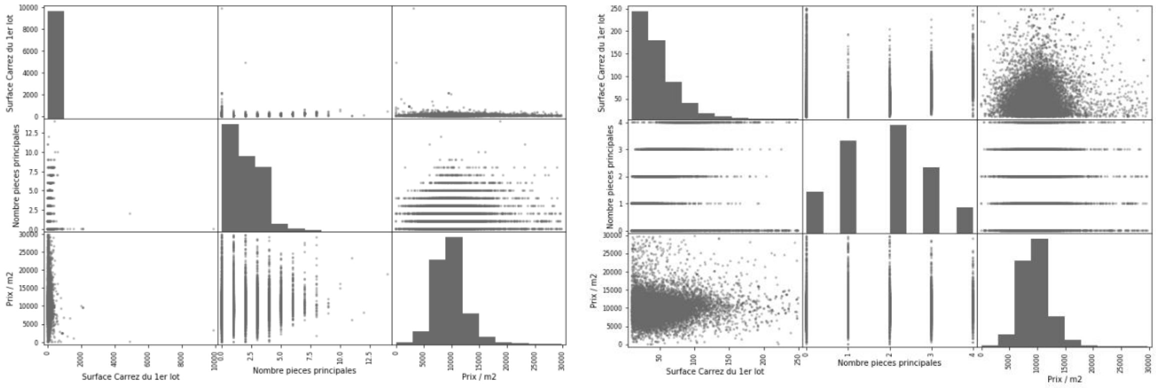
The fourth data source is Foursquare, accessible through its Python API, which will provide information about the location of popular venues in Paris on a *Quartier*-refined mesh. This will allow to identify 'hot spots' by category (family-friendly, clubs & bar, business, cultural, …) that will necessarily have an impact on the price of the transaction.

Methodology

Let's first clean the data. We remove all the data rows missing values, the filter out all the addresses that are not located in Paris. For that we simply use the postcode. We also remove all the unnecessary data columns that would complicate / slow down the subsequent computation processes without adding value.

We add a new column 'Prix / m2' which simply is the price at which the good as been sold, divided by its surface.Then, we remove all goods with exceptionally high price per square meter value (corresponding to luxury goods that are not the target of our study)

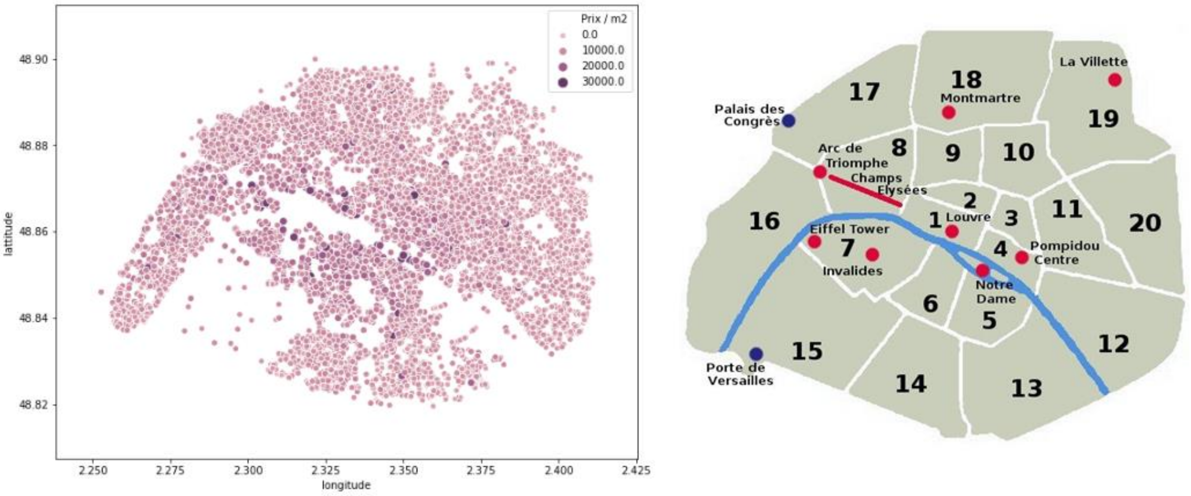
The following plot is the scatter matrix of the dataset. There are obvious outliers that deserve to be removed in order not to pollute the study. We use this chart to filter out these outliers to appropriate values, the result is a much nicer, balanced dataset.



Cleansed data before and after filtering outliers

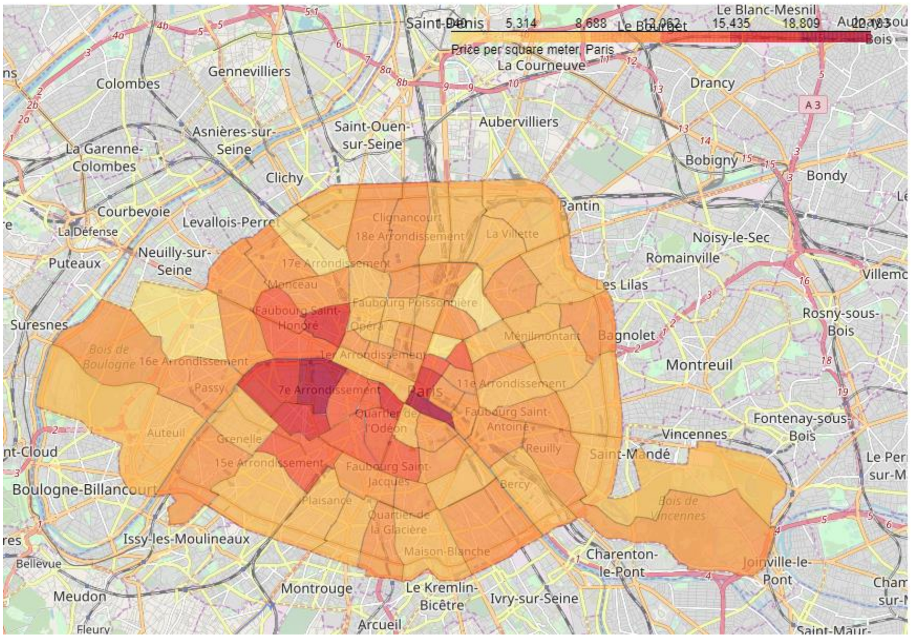
Each data point corresponds to a good that has been sold in 2018. They are referenced by address. We cannot use the address to properly localize a good, thus we need to use the openstreetmap API to get the latitude and longitude for each of the address of the filtered dataset.

This allows us to finally have a look at the mapping of the data over Paris area. What we can tell is that the pricy goods tend to locate around the Seine river and at the center of the city. We can also see that there are very few data point for the 15th district of Paris thus, we should be careful about the interpretation any model could yield about goods located in this district.



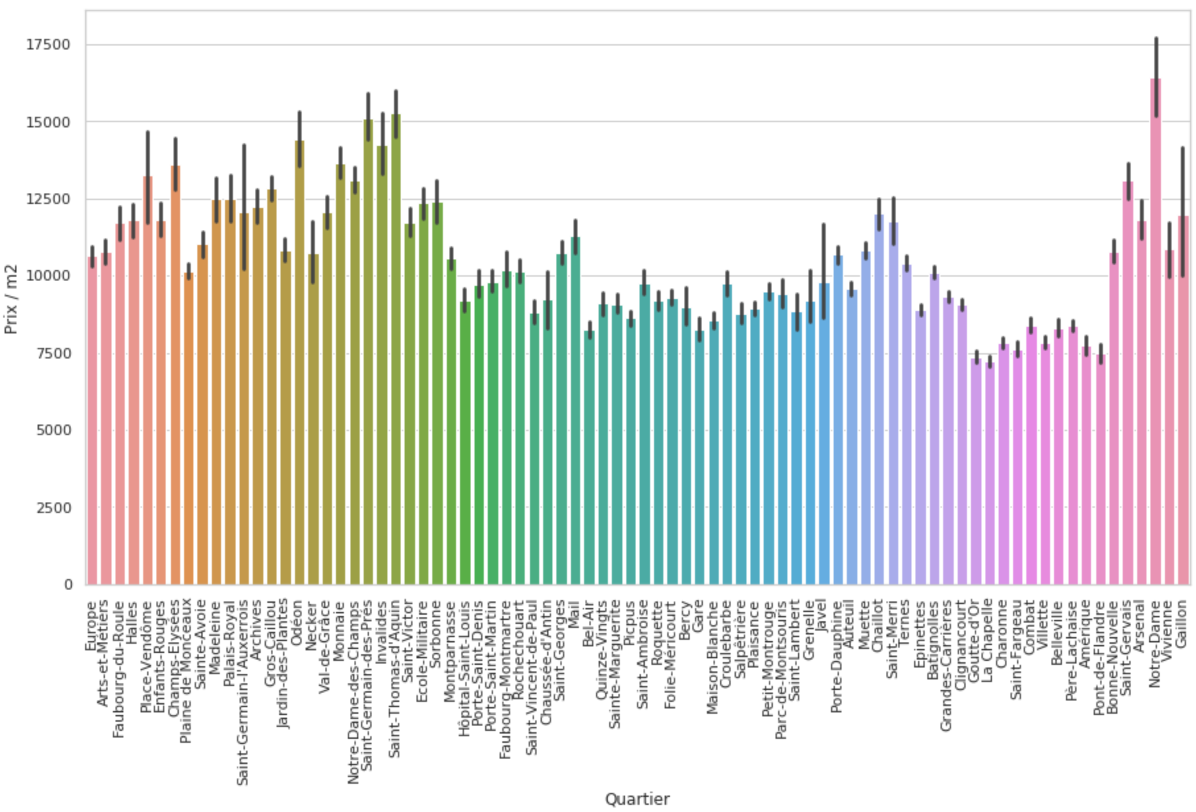
Distribution of goods with price category in Paris (2018)

We can also visualize the price per square meter distribution over Paris quartiers using a choropleth map:



Real estate price per square meter in Paris (2018)

Let’s now have a look at the average price distribution over quartiers to see if the starting hypothesis is justified. There are four quartiers per district, each next to another on the following chart. We can see that there are a lot of discrepancies in the average housing price per square meters for quartiers within the same district. The base assumption seems to be justified.



Real estate price per square meter per quartier in Paris (2018)

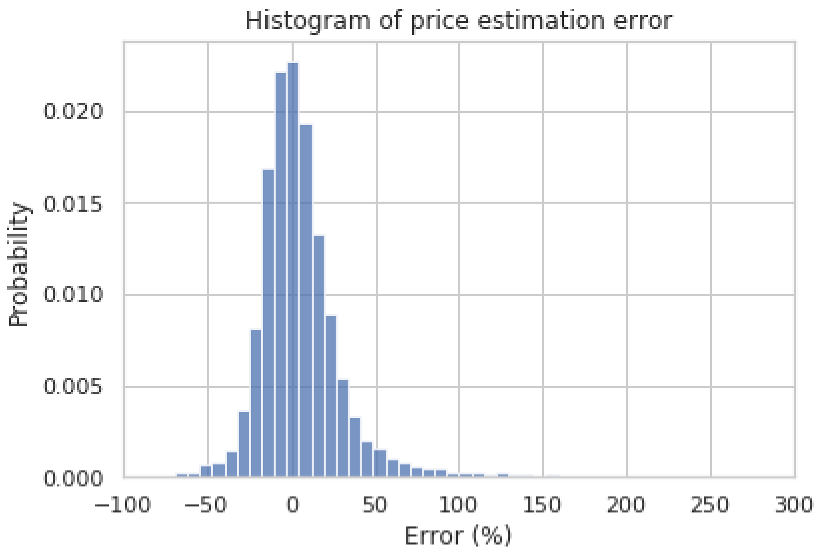
To predict the price of a good, we will first use a multidimensional linear regression. The model based on this regression shall be built using the following dependent variables: the surface of the good, the number of rooms and the location (using the name of the quartier). The target variable will obviously be the price per square meter.

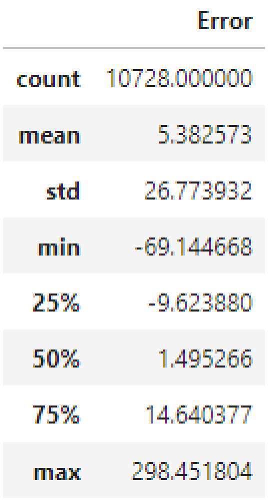
Before launching any algorithm, we must preprocess further the data, using the following pipeline:

- One hot encoding the quartiers in which the goods are located (as linear regression cannot deal with categorical variables)
- Generate a training set (that will also be used for validation of hyperparameters) and test set
- Normalize the surface and price values using standard scaling

Results

By first training the full dataset, and using the model to predict the input values, we end up with the following error distribution (in % of the actual price). The average error is around 5%, which is not bad.

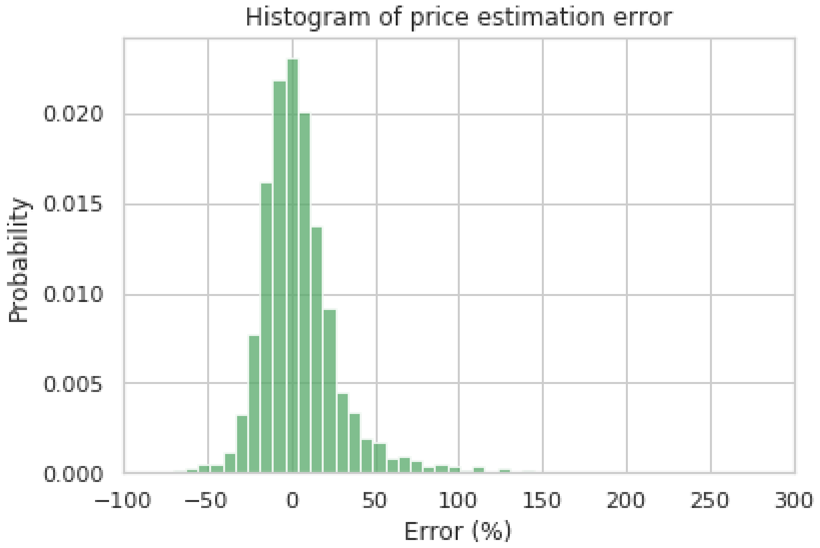


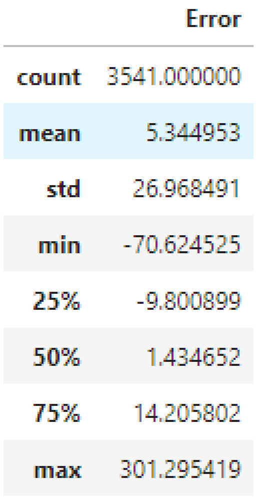


	Error
count	10728.000000
mean	5.382573
std	26.773932
min	-69.144668
25%	-9.623880
50%	1.495266
75%	14.640377
max	298.451804

Multiple linear regression over the whole dataset

Let’s now use a data split in order to verify that the model does not overfit the data. We can see that the results that we got with the train / test split are similar to the one we got from the whole dataset prediction. Hence it looks like this model is quite robust and could be deemed as acceptable.





	Error
count	3541.000000
mean	5.344953
std	26.968491
min	-70.624525
25%	-9.800899
50%	1.434652
75%	14.205802
max	301.295419

Multiple linear regression over a train/test split

Discussion

The results yield by this model clearly could help better assess the price of a good based on its precise location and some key characteristics. The model could be ameliorated using more parameters about the goods (presence of a lift, age of the building, floor, …).

This model could probably also be enhanced using the location data in order to build clusters of real estate goods; which would not depend on actual *Quartier* belonging but rather ‘unofficial’ neighborhood. This analysis could be conducted using a non-supervised method such as K-mean.

Conclusion

This captstone project allowed to develop a model for Paris real estate pricing estimation based the location of a good, its size and number of rooms using a multidimensional linear regression.