Network Tour of Data Science

Swiss Federal Institute of Technology (EPFL)
Master Phase - Fall 2019

Group n^{o}27

Airbnb price prediction from an host point of view

Florian Hartmann, Sylvain Lugeon, Paul Mosser, Samuel Furter

January 10, 2020

1 Introduction

Airbnb has become an easy way to find accommodations for more or less long stays in big cities all around the world. With the number of tourists constantly increasing, more people than ever before are using the Airbnb system. Instead of focusing on the travelers, we found interesting to guide attention towards accommodation providers.

"How much money can I ask for my own flat ?"

This is the question we will try to answer while focusing on the New York City Airbnb Open Data dataset. Indeed, Airbnb offers a lot of possibilities for the travelers to find the flat that suits their needs. Nevertheless, finding information about the price of your own flat depending on a precise area might be more difficult. Providing information to the accommodation provider is the aim of the product we developed.

The chosen dataset provides different information about every Airbnb of New York. The location, the number of accommodations per host, the price, the availability as well as the type of room are relevant numbers and informations that help us to elaborate an efficient tool to help people that want to jump in the venture Airbnb. Indeed, we should be able to predict the price using the information cited before. To do so, we will use machine learning techniques enhanced by a graph structure formed by the Airbnbs.

2 Acquisition

The dataset is directly provided by Airbnb and is very clean, each entry represents an Airbnb accommodation. The first step is the construction of the graph. It is a natural choice to represent the accommodations as the nodes. The edges can then depend on the distance between the accommodations. The graph should represent the geographical distribution of the accommodations. Each nodes will have some associated features, as cited before.

2.1 Creating the edges

The problem is that the features describing the location of the accommodation is based on the *WGS84* mode, i.e in *latitude* and *longitude* coordinates. Those coordinates are not very suited to construct our graph, as they describe a position on a sphere. Another approach would be to use Universal Transverse Mercator (UTM) model, that consider small areas of the Earth as flat and describe positions as plane coordinates. As a side effect, this model is less accurate. But because the area we are looking at, i.e the city of New York, is quite small, the loss in accuracy is very small. After transforming the *WGS84* coordinates into UTM coordinates, we can easily compute the euclidean distance between accommodations and construct the edges. Moreover, as UTM coordinates describe a plane, it allows use visualize the geographical structure of the accommodation while plotting the graph.

2.2 Sampling

The dataset contains 48’895 Airbnbs localized in the city of New York. To avoid computational issues we only keep 5’000 of them with a random sampling. Nevertheless, in the last section we’ll explore one of our model using more samples.

2.3 Removing outliers

A quick look at the price distribution within the dataset shows a lot of outliers. This is explains by the nature of some Airbnbs, it is note rare to find very luxurious accommodations at significant price while browsing the Airbnb search engine. As ML models are usually not good at dealing with outliers and because the aim of this project focus on *lambda* people that want to rent their accommodation, we decide to only keep the entries that have a price below the 0.95 quantile of the price distribution.

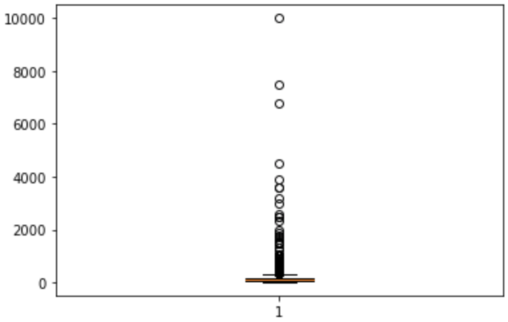


Figure 1: Price boxplot

3 Exploration

Now that we have the euclidean distance between all the nodes, we can construct our graph. First, we construct a **RBF-kernel based graph**, where the weight of an edge is given by:

$$w(\mathbf{x}, \mathbf{x'}) = exp^{-\frac{\|\mathbf{x}-\mathbf{x'}\|^2}{2\sigma^2}} \tag{1}$$

The parameter σ is chosen according to the mean distance between the nodes, and we then use a threshold ϵ to sparsify the matrix, i.e $w(\mathbf{x}, \mathbf{x'}) < \epsilon = 0$. Parameters σ and ϵ have been chosen such to create a connected graph, but with a sparse adjacency matrix. Using again the euclidean distance, we can also construct a **k-nearest neighbors** graph, where two nodes are connected if one of them is part of the k nearest neighbors of the other. We will explore that kind of graph with $K = 50$, $K = 200$, $K = 400$.

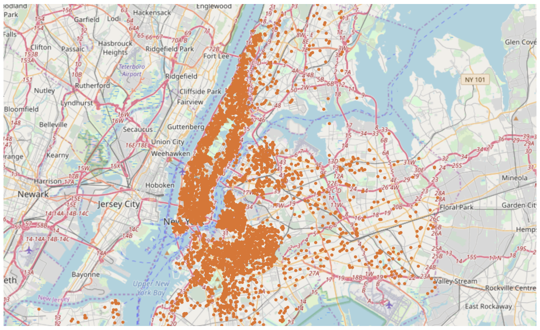


Figure 2: New York City map with Airbnb

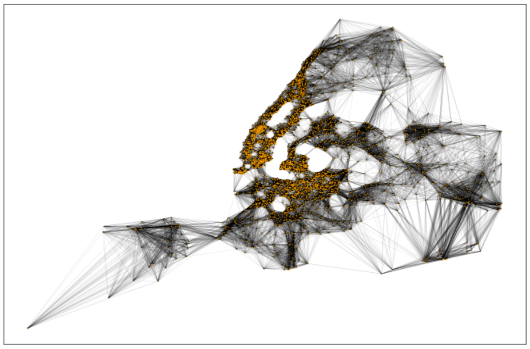


Figure 3: Graph Structure from k-NN 50

3.1 Graphs characteristics

Figures 4 and 5 shows plot the adjacency matrix of the RBF-kernel based graph and the 200-NN graph and table 1 shows basics characteristics. From a quick look at these, we can tell the the two kinds of graphs are very different. It will allows us to improve our ML model using two very different graph structures, and determine which graph lead to a better improvement.

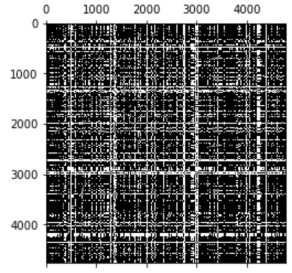


Figure 4: Adjacency matrix with RBF kernel

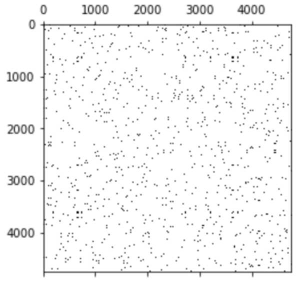


Figure 5: Adjacency matrix with KNN-200

For KNN graphs, the average degree is higher than the chosen number of neighbors. This is because if we take two nodes A and B, B might not be one of the k-nearest neighbors of A but it is possible that A is one of the nearest neighbor of B. In that case, an edge is still created between the nodes.

model	**average degree**	**edges**	**connectivity**
RBF kernel	3422.27	weighted	connected
KNN-50	59.20	unweighted	connected
KNN-200	243.95	unweighted	connected
KNN-400	495.84	unweighted	connected

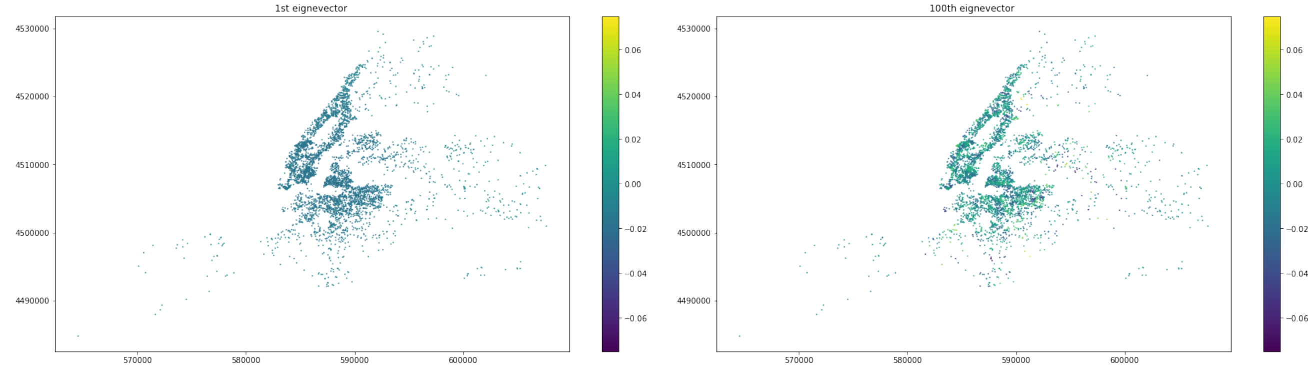
Table 1: Characteristics of the graphs

3.2 Spectral components

As we will use the spectral component of the graphs to improve the ML model, it is meaningful to explore the spectral decomposition of the graphs laplacian. Note here that we used a *normalized* laplacian. Figure 6 shows the 1st and the 100th eigenvectors of RBF-kernel based graph. As we could expect, the 1st eigenvector is way smoother than the 100th.

4 Exploitation

We are now interested in predicting the price of a new Airbnb, based on features known to a new host. We'll first follow a baseline approach where we'll feed raw features to a Machine Learning model. One of the main problem is that of all the features in the dataframe, we are only allowed to use the ones that are known to a new accommodation, i.e we can't use features such as the *availability* or the *number of reviews per month*. The ML model we use is a ridge regression model



(a) 1st eigenvector

(b) 100th eigenvector

Figure 6: Difference of smoothness between 1st and 100th eignvector

with 5-fold cross-validation and grid-searched regularization parameter (note that it is a quite simple model). We choose MSE to assert the precision of our model.
The used features are the following:

- The neighborhood (Manhattan, Brooklyn', Queens, Bronx' or Staten Island)
- The number of accommodations per host
- The type of accommodation (home, private room or shared room)

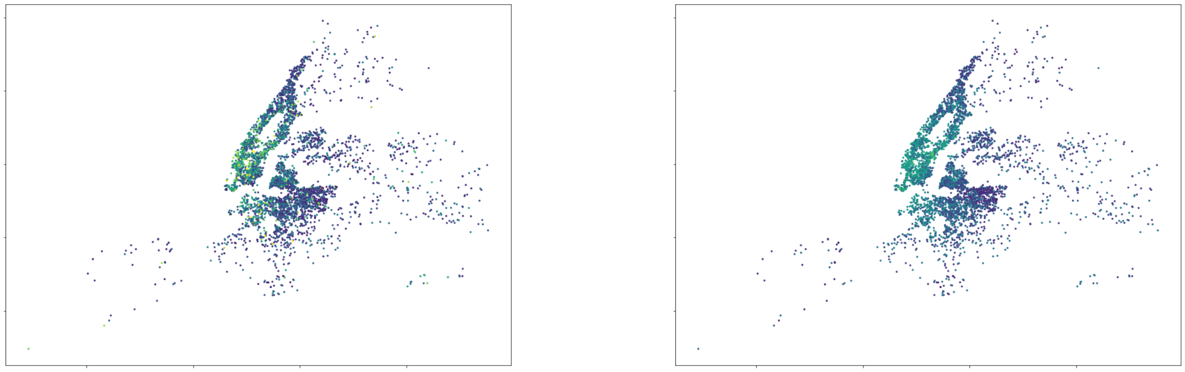
The baseline approach led to a MSE of **2950.83**. We will now use the graph structure to improve that result. The assumption is that close accommodations in the graph should have similar price, thus we'll filter the features with an Tikhonov regularization filter (2) that reduce the high-frequencies contribution. The obtained results are shown in the table 2. The MSE is computed from the predictions of the whole samples set, we have to use the same samples we constructed the graph from. The Tikhonov regularization filter is a function of the eigenvalues of the graph laplacian and directly depends on the largest eigenvalue. The parameter c allows allows us to scale the filter, so that high-frequencies have less (or more) weights. We used there the value $c = 3$, as it led to the best results.

$$tk(e) = 1/(1 + \alpha * e) \text{ with } \alpha = c * 0.99/e_{max} \tag{2}$$

Model	**MSE**	**MSE using graph structure**	**Improvement**
RBF kernel	2950.83	2864.68	2.92%
KNN-50	2950.83	2801.92	5.05%
KNN-200	2950.83	2686.21	8.97%
KNN-400	2950.83	2705.95	8.30%
KNN-200 (20'000 s.)	2950.90	2806.96	4.88%

Table 2: MSE of various models and price prediction improvement

The last line of the table 2 plots the results of the best model, but computed on 20’000 samples (and not 5’000 as for the others models). The Figure 7 shows the difference between the groundtruth price and the predicted price using the KNN-200 graph.



(a) Groundtruth

(b) Prediction (k-NN 200)

Figure 7: Difference between groundtruth and prediction using the graph structure

5 Conclusion

The goal of this project was to estimate the price an accommodation provider could ask for his flat on Airbnb. In order to have a consistent estimation we only used the features a new provider would be able to give when signing in Airbnb. The baseline approach was to only use a ML model on those features, and then use a different graph structures to improve the predictions. This led to an improvement of the estimation with all the graph structures. The graph that improves the most the prediction is the 200-KNN graph. But using more samples do not really improve the prediction.

Despite we saw an improvements over the prediction using the graphs, we are still far from the real price a flat could be rent. It is explained by the fact that we restricted ourselves by considering only the features a new host could dispose of, so important features like the *availability* or the *number of reviews per months* could not be used in the predictions. Moreover, some features were probably missing in the dataset, such as the size of the flat. Indeed, having this information could really improve our prediction. Considering such new features could be considered as a further improvement. Another improvement could also be to add a user interface, with the possibility to enter a position either by hand or by clicking on a map (possibility to do it with *folium*) and to have a personalized output.

References

[1] New York City Airbnb Open Data,
`https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data`