The battle of neighborhoods: finding a place for your gym

Final report

SEPTERMBER 2019

Authored by Gabriele Favia

coursera

IBM®

Table of the contents

Introduction

Using the data to know what to do

In the previous decade, the explosive diffusion of data management and retrieval system developed numerous applications of “Business Intelligence” and “Decision Support Systems”: software born to help the business owners to make effective choices based on past data digested in a statistical report form.

Nowadays, machine learning research is continuously rising the level of reliability of the previsions, lowering at the same time the costs of these systems.
Indeed, a series of factors such as the spreading of high-speed internet connection, bigger data storage capacity and improved computational performance is leading to massive amount of data, also accessible in public form, that are super-charging the robustness of the models.

> ***“It is a capital mistake to theorize before one has data.”***
> ***Sherlock Holmes, A Study in Scarlett (Arthur Conan Doyle).***

This document will explain how to navigate through the sea of available data in the bay of Foursquare to get useful insights on what place can be the best one to open a gym in an Italian medium-sized city such as Bari.

Definition of the problem

Where to place your business?

In a world where open a new business is pressed by the risk of failure due to bad management decision, let's take a step back and see the head of the chain of good business practices: choosing the right place.

Given a city and a possible type of shop/business/POI, is it possible to estimate which one of the vacant premises is the most suitable for a new opening?

Of course, the quantity of variables to make a solid decision is unconceivable, but for a practicing purpose the idea is looking at the other gyms in similar cities to the target one, extracting the ratings given by the clients, and put them in correlation with the quantity of POIs/venues in their surroundings.

In the end it should be possible to estimate, for pure fun, which one of the neighborhoods composing the target city is the most suitable to host the next gym.

Although the focus here is on the specific case of "gym" typology and "Bari" city, the resulting model can be rebuilt by modifying some parameters in the code.

This model can, therefore, give an insight about the next suitable place for a business activity, looking only at the correlation between successful gyms and the distribution of other POIs/venues in the surrounding area.

Finding data sources

Being able to know where to search

As known, machine learning algorithms can start to train a model only with a certain amount of data.

One of the possibilities to get open data, is using the APIs released by Foursquare. Foursquare is a social reviewing web platform which aggregates information about venues/POIs around the world, with the contribution of its users.
The access to information is possible through RESTful paradigm requests which can give back different data, depending on the endpoint; the used ones are the following:

Name of the endpoint	Description
search	Returns a list of venues, given a basic coordinate and a maxim radius, or alternatively a 'near' reference.
venue	Returns all of information related to a single venue, given its ID.

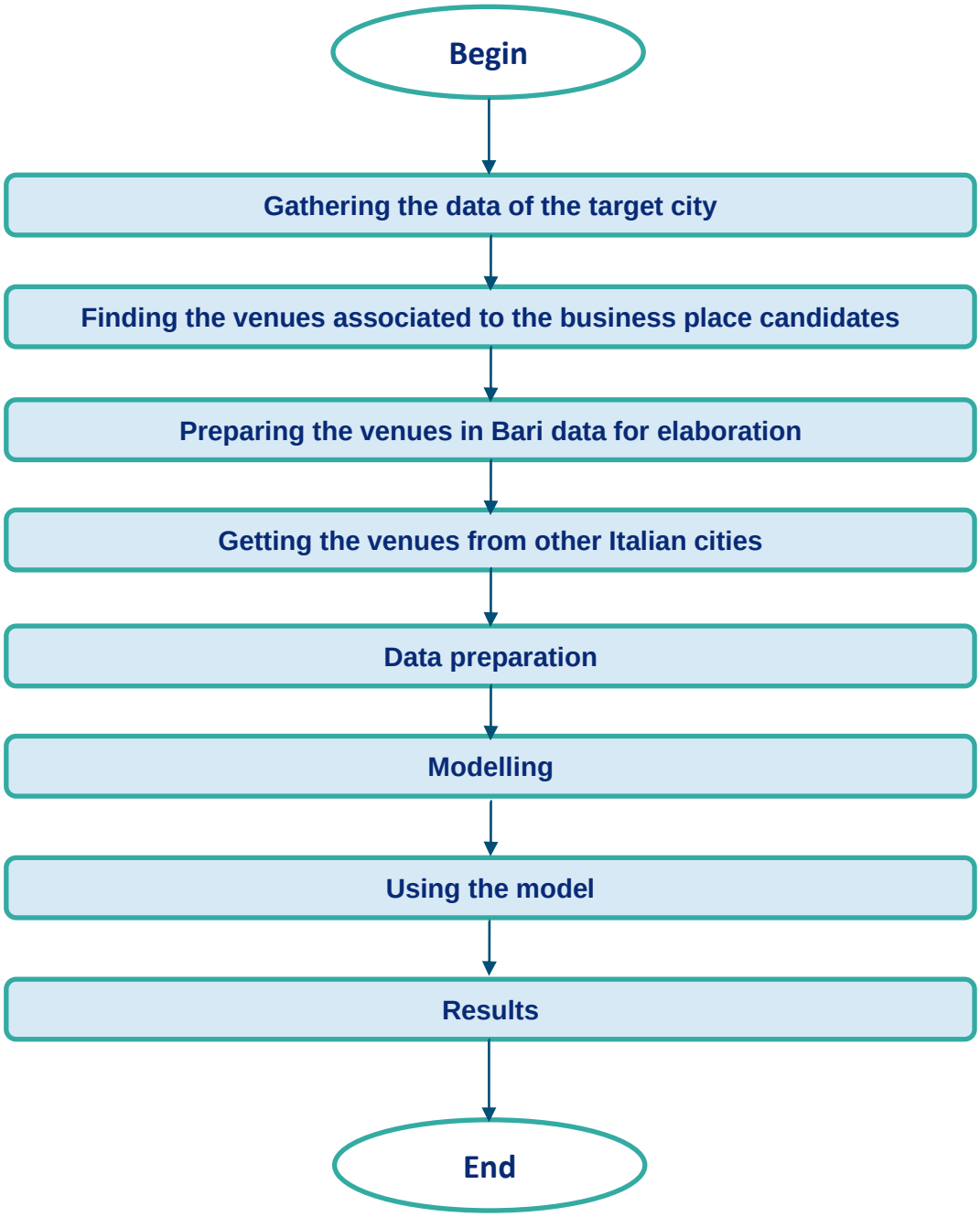
- The *search* endpoint has been used to get all the POIs around the business candidates in the target city and to get all the POIs around the gyms in the other Italian cites.
  Each POI found is listed by its category (Wine Bar, Dessert Shop, Pharmacy etc.)
- The *venue* endpoint in combination with a venue ID, endpoint returns all the details about the selected POI, including the rating, which will be used to classify "Not Suitable" gyms from "Suitable" gyms.

Data relative to the places for rent in Bari, are acquired from a popular Italian website (casa.it) making parametric https calls and treating the results using webscrapping.
The request to casa.it involves the NW and SE geographical boundaries, the type of search (business or house), the surface in meters and if the target is for rent or for buying as inputs. In this particular case the boundaries are related to Bari area (retrieved from Google Earth), 400 square meters of extension and rent mode.

Information about the position of Bari neighborhood will be provided by Wikipedia using webscrapping at https://it.wikipedia.org/wiki/Quartieri_di_Bari and the map of the boroughs by converting the shapefile from the official website of the Bari Municipality http://opendata.comune.bari.it/dataset/circoscrizioni into a geojson file to overlay on a folium instance .

Workflow

A step-by-step procedure to solve the problem



Methodology

Defining the attack-plan

- **Gathering the data of the target city**: for the target city, in this case Bari, has been conducted a research about the position of the vacant business places on a Italian real estate listing website (casa.it), parametrizing the query with the boundaries of the area and the amount of surface needed.
  At the same time, using data from Wikipedia.it, has been possible to get the positions and the names of the neighborhoods composing Bari.
  All this data is then overlapped with a shape file of the boroughs on a folium map (served by OpenStreetMap).
- **Finding the venues associated to the business place candidates:** for each one of the vacant places found, Foursquare provided all the venues/Points of Interest inside a predefined radius.
- **Preparing the venues in Bari data for elaboration:** grouping the venues associated with the places found in Bari after using the one hot encoding technique provided the data binding between features and the candidates.
- **Getting the venues from other Italian cities**: given a range of population, the algorithm searches into certain Italian cities the gyms present in the Foursquare database. For each resulting gym, only the ones with a rating are kept and as happened to the candidate one, a deeper analysis and listing of surrounding venues has been done, following once again by one-hot-encoding and grouping.
- **Data preparation:** in this section the features being part of candidates features and city-gyms features are intersected. Without this step, the model can't work on the final business place candidates of Bari because it expects the same number of features present in the ultimate training dataset.
- **Modelling:** In this case, the clue of which place is better is the output of a supervised machine learning algorithm, where the suggested places to set a new business should have the same surrounding situation of the gyms with rating above a certain threshold.

The supervised algorithms chosen are Logistic Regression, Support Vector Machines and K-Nearest Neighborhood, Decision Tree and Naïve Bayes using the Scikit-learn library for Python.
In a so small dataset has been chosen to use the 75% or entries for training, leaving the remaining 25% to the testing.

- Logic Regression: all the available solvers (liblinear, newton-cg, lbfgs, sag, saga) have been tested giving the same score results.
  At first, the finding of true positives where completely unbalanced towards the “Not Suitable” with a score of 71%, while “Suitable” were recognized with a precision of 0%. Unacceptable.
  By playing with the “C” parameter, it has been possible to find a 58% – 17% balance (fig 1) setting it to the value 8.
  Due to the low result, continuing the research with other algorithms has been unavoidable.

```
Classification report with  liblinear  solver:
              precision    recall  f1-score   support

          No       0.50      0.70      0.58        10
         Yes       0.25      0.12      0.17         8

   micro avg       0.44      0.44      0.44        18
   macro avg       0.38      0.41      0.38        18
weighted avg       0.39      0.44      0.40        18

Classification report with  newton-cg  solver:
              precision    recall  f1-score   support

          No       0.50      0.70      0.58        10
         Yes       0.25      0.12      0.17         8

   micro avg       0.44      0.44      0.44        18
   macro avg       0.38      0.41      0.38        18
weighted avg       0.39      0.44      0.40        18

Classification report with  lbfgs  solver:
              precision    recall  f1-score   support

          No       0.50      0.70      0.58        10
         Yes       0.25      0.12      0.17         8

   micro avg       0.44      0.44      0.44        18
   macro avg       0.38      0.41      0.38        18
weighted avg       0.39      0.44      0.40        18
```

Classification report with sag solver:

	precision	recall	f1-score	support
No	0.50	0.70	0.58	10
Yes	0.25	0.12	0.17	8
micro avg	0.44	0.44	0.44	18
macro avg	0.38	0.41	0.38	18
weighted avg	0.39	0.44	0.40	18

Classification report with saga solver:

	precision	recall	f1-score	support
No	0.50	0.70	0.58	10
Yes	0.25	0.12	0.17	8
micro avg	0.44	0.44	0.44	18
macro avg	0.38	0.41	0.38	18
weighted avg	0.39	0.44	0.40	18

Figure 1 – Scores with logistic regression

- SVM: the principal kernels *(linear, poly, rbf, sigmoid)* have been used to find out the one with the best results. In all the cases, probably to the to much small dataset, the results were the same using the default value of the "*C*" parameter (1.00). Rising the *C* parameter till 8, the differentiation between kernel types became consistent, but still, where the avg f1-value was increasing at the same time the precision to find true positives suffered unbalance towards the "No" label, making these models to reliable (fig 2).

Classification report with linear kernel:

	precision	recall	f1-score	support
No	0.42	0.50	0.45	10
Yes	0.17	0.12	0.14	8
micro avg	0.33	0.33	0.33	18
macro avg	0.29	0.31	0.30	18
weighted avg	0.31	0.33	0.32	18

Classification report with poly kernel:

	precision	recall	f1-score	support
No	0.46	0.60	0.52	10
Yes	0.20	0.12	0.15	8
micro avg	0.39	0.39	0.39	18
macro avg	0.33	0.36	0.34	18
weighted avg	0.35	0.39	0.36	18

Classification report with rbf kernel:

	precision	recall	f1-score	support
No	0.47	0.70	0.56	10
Yes	0.00	0.00	0.00	8
micro avg	0.39	0.39	0.39	18
macro avg	0.23	0.35	0.28	18
weighted avg	0.26	0.39	0.31	18

Classification report with sigmoid kernel:

	precision	recall	f1-score	support
No	0.50	0.80	0.62	10
Yes	0.00	0.00	0.00	8
micro avg	0.44	0.44	0.44	18
macro avg	0.25	0.40	0.31	18
weighted avg	0.28	0.44	0.34	18

Figure 2 – Scores with support vector machines

- K-Nearest Neighbors: by overriding the default value of numbers of neighbors (5) with different integers from 2 to 7, the best score turned out to be 3 (fig 3). So far, KNN provided the best score among the ML algorithm tried for this dataset.

	precision	recall	f1-score	support
No	0.62	0.80	0.70	10
Yes	0.60	0.38	0.46	8
micro avg	0.61	0.61	0.61	18
macro avg	0.61	0.59	0.58	18
weighted avg	0.61	0.61	0.59	18

Figure 3 – Scores with KNN

- Decision Tree: the algorithm run using both *Gini* and *Entropy* criterion and various values for the parameter min_samples_split (default = 2), till the optimum among the combinations tried has been found to be *Gini* and *0.2*, giving almost the same avg f-1 score in comparison to

KNN but a less uniform precision between the two label to predict (fig 4).

	precision	recall	f1-score	support
No	0.64	0.70	0.67	10
Yes	0.57	0.50	0.53	8
micro avg	0.61	0.61	0.61	18
macro avg	0.60	0.60	0.60	18
weighted avg	0.61	0.61	0.61	18

Figure 4 – Scores with Decision Tree

- Naïve Bayes: to complete the comparison between the most popular algorithms, Naïve Bayes has been used, giving as expected, not-so-impressive results. Sklearn doesn't provide any interesting parameter to tune for this technique.

	precision	recall	f1-score	support
No	0.43	0.30	0.35	10
Yes	0.36	0.50	0.42	8
micro avg	0.39	0.39	0.39	18
macro avg	0.40	0.40	0.39	18
weighted avg	0.40	0.39	0.38	18

Figure 5 – Scores with Naïve Bayes

In the end, the KNN has been chosen because of the good results in terms of avg f1-score and precision in comparison to the other models, although results are not objectively ideal.

A more accurate evaluation of the model has been done with tool visual tools, like the confusion matrix provided by the seaborn library for Python.
As it's possible to see in the (fig 6), the true positives are effectively recognized in the case "No" label ('Not Suitable' here represented by 0) while the same recognition strength can't be found in the true positive related to the "Yes" label ('Suitable' here depicted by 1).

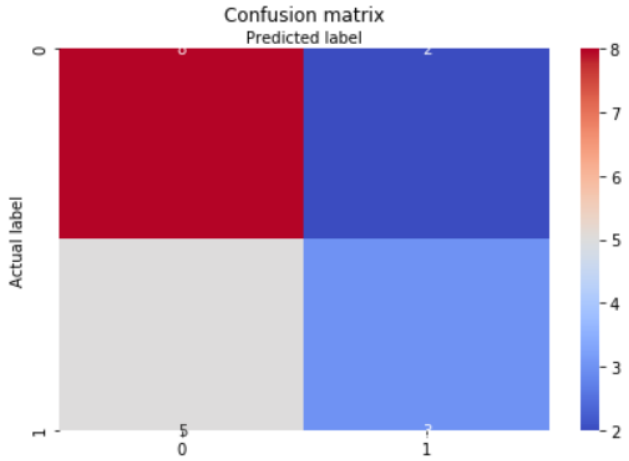


Figure 6 - Confusion matrix for the KNN model

One more verification has been provided by tests done with the K-Folds technique (fig 7). The simple average of the seven cycles is 0.5348 circa which is results still weak.

```
K-Nearest Neighbor with 2 folds -> score: 0.5769841269841269
K-Nearest Neighbor with 3 folds -> score: 0.4939613526570048
K-Nearest Neighbor with 4 folds -> score: 0.4934640522875817
K-Nearest Neighbor with 5 folds -> score: 0.5342857142857144
K-Nearest Neighbor with 6 folds -> score: 0.5631313131313131
K-Nearest Neighbor with 8 folds -> score: 0.5503472222222222
```

Figure 7 - K-Folds results for each cycle

- **Using the model:** once the right algorithm with its parameters has been found, the model has been trained again with all the rows related to the gyms (no splits), and then it's been applied to the dataset of the free business estates of Bari.
  In the (fig 8), on a total of 14 gyms, 3 were catalogued as suitable while 11 as not suitable.

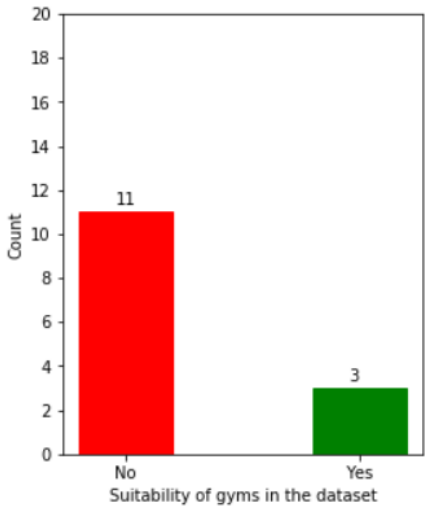


Figure 8 - Suitability of business estates found in Bari

Results

Identifying the solution

In the battle of neighborhoods, the one with more suitable places shall be the winner; a function made up to calculate the corresponding neighborhood based on geometric distance assigned each place found to its belonging neighborhood, resulting in a uniform distribution among three neighborhoods: Marconi-San Girolamo-Fesca, Stanic-Villaggio dei lavoratori and Picone (fig 9, 10).

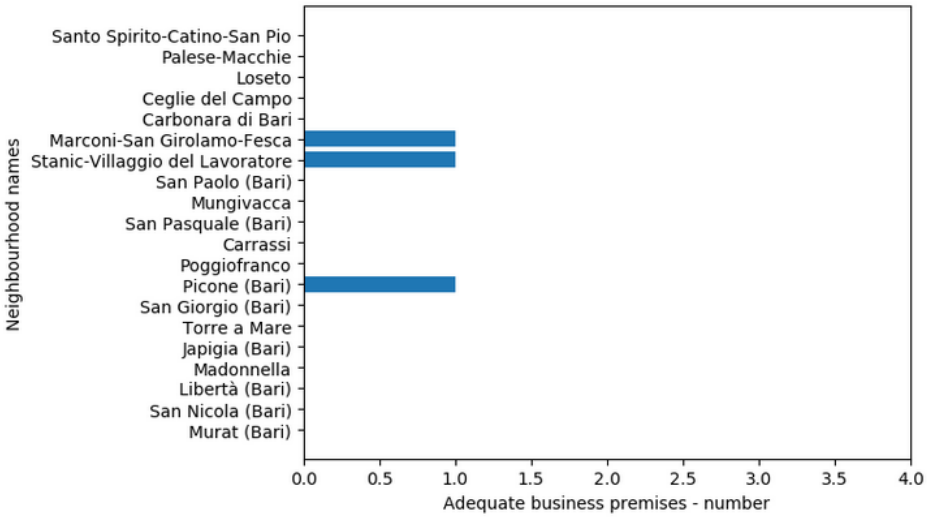


Figure 9 - Number of suitable places for neighborhood

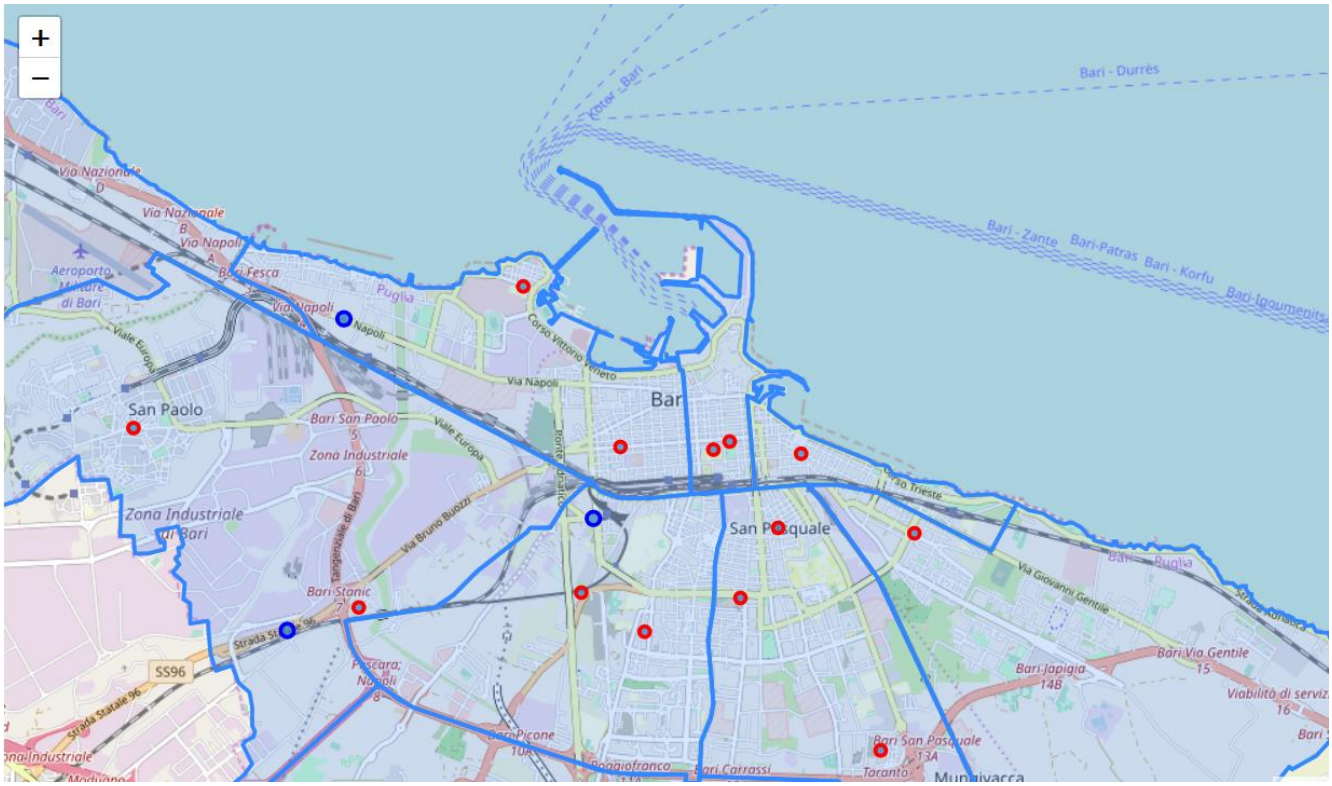


Figure 10 - Map with neighborhood (red circles) and places found (blue circles), polygons are delimiting boroughs

In this small datascience research, three neighbourhoods won "the battle" of having a probable business premise for a gym in the city of Bari, Italy; these are: "Marconi-San Girolamo-Fesca", "Stanic-Villaggio del Lavoratore" and "Picone".

In absence of other known factors, this small model can give a hand in choosing the right place for a cold-start business.

Discussion

Limitations and future developments

- The first and most important limitation is the assumption that the rating of a gym can be discriminated only by the surrounding businesses. That is of course not true but this assumption can be taken in account for future development as a subset of feature to analyze.
- The whole research of venues is limited by the Foursquare API number of calls per day, therefore the LIMIT parameter of venues around each gym has been limited to 100 entries, in fact the free basic account of Foursquare was not enough, so it’s been necessary to switch to the upper category where the platforms requests the credit card number.
- The parameter RADIUS used for the search of venues is set to 250 meters, this is considered the maximum limit of distance in which a person is willing to walk before or after the gym session. No information in the social sciences literature has been found about it.
- The cities to train with have population between 180000 and 420000 units: this choice has been made looking at the calls limit in Foursquare API and because the model wanted to take in account the behavior of an average city inhabitant in Italy.
- The threshold rating imposed to 7.0 is the result of average quality perception of a point of interest.
- The host used to search for free location in Bari is casa.it, which data are scrapped from a single webpage. An extensive system should be able to use API calls or at least being able to scrap multiple websites being aware that the possible clones of announces should be deleted in the phase of data preparation.
- The boroughs of Bari are following the old grouping system: that's because the new one (with "Municipi") tends to group too many neighbourhoods together, making the differentiation of zones too much simplistic.

References

Bibliography and more

Hu, Longke & Sun, Aixin & Liu, Yong. (2014). Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. 10.1145/2600428.2609593.