IBM Applied Data Science Capstone Project

Predicting house prices in Monza using Foursquare API

Alessandro Bonvini, 2021 February 11th

1. Introduction: business problem

The goal of this project is to study the impact of using Foursquare API on the regression problem of predicting house prices in the Italian city of Monza.

The possible advantage of using Foursquare API is related to the fact that the square meter price of the houses in Monza, like in most of cities, varies depending on the position of the house.

The presence of one or more important monument, church or square or famous restaurants, shopping centers etc. close to a house, in fact, will likely have an impact to its square meter price.

In the project I will check if the information given by Foursquare API could be used to retrieve spatial information that can be used by different regression models.

In particular, we will check if the information given by Foursquare API could be used to automatically cluster the houses in trending neighborhoods and then use the obtained cluster information as an additional feature used by the machine learning models.

Alternatively, we will explore if adding the top trending venues directly as features to the models will lead to a better performance with respect to using the cluster information or no spatial information at all.

I will test 4 regression models: Multivariate Linear Regression, Ridge Regression with 3rd degree Polynomial Features, SVR and Random Forest.

1.1 Interested audience

The house prices prediction can be a useful feature to be implemented in houses selling announcements search engines.

With this feature implemented, in fact, the seller can have an idea of the reasonable value of the house that he wants to sell and he can be warned if the price of the announcement is too far from the fair price.

At the same time, the buyer can understand what a good price for the house that he wants to buy could be, obtaining an indication about the price fairness.

2. Data

2.1 Initial dataset

After searching for available data sources that could be used for the project, I realized that no houses databases are public available.

Therefore, with the help of an estate agent working in Monza, we selected what could be the most impacting features to the house prices and we extracted hundreds of houses from his management system database, compiling afterwards a set of representative features for each house.

In particular, we created a csv dataset of 405 houses to be used as training set, and another csv dataset of 44 houses to be used as test set to evaluate the generalization performance of the various models.

We decided each house to have the following features:

1. **PRICE** – the price of the house, in Euros.

2. **ADDRESS** – the street of the house, in the following format: *Number Street.*

3. **ROOMS** – the number of rooms of the house. Bathrooms are not considered as rooms.

4. **METERS** – the commercial square meters of the house.

5. **BATHROOMS** – the number of bathrooms of the house.

6. **FLOOR** – this feature describe the main floor of the house. Possible values are:
   a. **GROUND**: the main floor of the house is at ground level
   b. **MIDDLE**: the main floor is between the first and the last floors
   c. **LAST**: the main floor is the last, so it is an attic or a mansard
   d. **VILLA**: the house is an independent villa

7. **FLOORS** – the number of floors of the house.

8. **YEAR** – the construction year of the house.

9. **STATUS** – the current conditions of the house. This is the most subjective and difficult to determine feature, but also one of the possible most impacting features in estate agent's opinion. Possible values are:
   a. **BAD**: the house needs to be renovated before being inhabited
   b. **GOOD**: the house don't need renovation to be inhabited
   c. **RENOVATED**: the house has been completely renovated in one of the last 3 years and it is in close-to-new conditions
   d. **NEW**: the house is a new construction

10. **TERRACE** – YES if the house has a terrace large enough to be used for eating, otherwise NO

11. **GARDEN** – YES if the house has a garden, private or common, that can be used to let the kids play, otherwise NO.

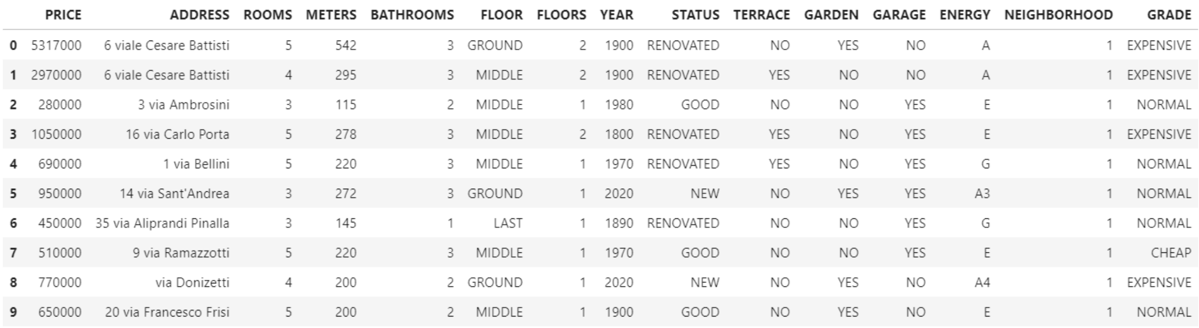
12. **GARAGE** – YES if the house has a covered place to be used for parking one or more cars, otherwise NO.

13. **ENERGY** – the certified energy class of the house. Possible values range from G, which is the lowest class, to A4 which is the highest class.

14. **NEIGHBORHOOD** – the neighborhood of the house. Possible values range from 1 to 9.

15. **GRADE** – The estate agent evaluation for the house price. Possible values:
    a. **CHEAP**: the house price is lower than the real value.
    b. **NORMAL:** the house price is average for the house value
    c. **EXPENSIVE**: the house price is too high.

Here is how the first 10 rows look like after reading with pandas:



	PRICE	ADDRESS	ROOMS	METERS	BATHROOMS	FLOOR	FLOORS	YEAR	STATUS	TERRACE	GARDEN	GARAGE	ENERGY	NEIGHBORHOOD	GRADE
0	5317000	6 viale Cesare Battisti	5	542	3	GROUND	2	1900	RENOVATED	NO	YES	NO	A	1	EXPENSIVE
1	2970000	6 viale Cesare Battisti	4	295	3	MIDDLE	2	1900	RENOVATED	YES	NO	NO	A	1	EXPENSIVE
2	280000	3 via Ambrosini	3	115	2	MIDDLE	1	1980	GOOD	NO	NO	YES	E	1	NORMAL
3	1050000	16 via Carlo Porta	5	278	3	MIDDLE	2	1800	RENOVATED	YES	NO	YES	E	1	EXPENSIVE
4	690000	1 via Bellini	5	220	3	MIDDLE	1	1970	RENOVATED	YES	NO	YES	G	1	NORMAL
5	950000	14 via Sant'Andrea	3	272	3	GROUND	1	2020	NEW	NO	YES	YES	A3	1	NORMAL
6	450000	35 via Aliprandi Pinalla	3	145	1	LAST	1	1890	RENOVATED	NO	NO	YES	G	1	NORMAL
7	510000	9 via Ramazzotti	5	220	3	MIDDLE	1	1970	GOOD	NO	NO	YES	E	1	CHEAP
8	770000	via Donizetti	4	200	2	GROUND	1	2020	NEW	NO	YES	NO	A4	1	EXPENSIVE
9	650000	20 via Francesco Frisi	5	200	2	MIDDLE	1	1900	GOOD	NO	YES	NO	E	1	NORMAL

Figure 1 - training houses dataframe

The houses have been chosen to be in the same number for each neighborhood, that is 45 for each one.

2.3 Adding meter price information and removing outliers

It is known that the price of a house is computed by multiplying the square meters by the square meter price of the house area.

Since this relationship is known, it makes sense to create a new target variable: the square meter price. In this way the models will be trained to predict this variable instead of the price, removing the known relationship between price and square meters form the problem.

For this reason, in addition to the initially chosen features, I decided to add the square meter price information, which will be used as target variable for the regression models.

After an initial visual analysis, which will be described in the following sections, I found out that two big outliers were present in the initial dataset.

The two outliers have been removed from the dataframe.

2.2 Adding spatial information and Foursquare API

Spatial information is expected to have a key role in improving the performances of the regression models.

By means of the address information, I used geopy library to retrieve latitudes and longitudes coordinates for each house.

I used ArcGIS as geolocator, because Nominatim failed to localize the house numbers in the city of Monza.

I then added latitudes and longitudes to the initial dataframe.

Regarding the usage of Foursquare API, the idea is to find what are the top trending venues in Monza, by exploring the area using Monza's coordinates and a radius of 2500.

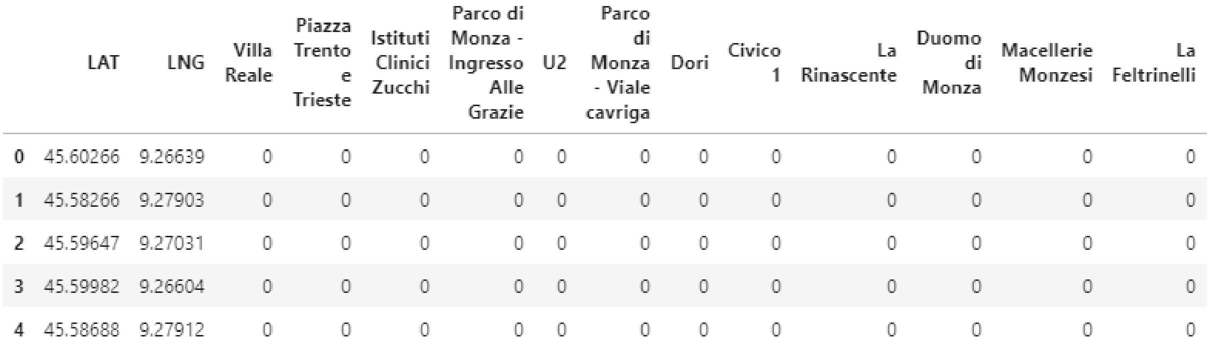
The search has been performed with a GET request to the *explore* Foursquare Endpoint, by specifying, in addition to the standard parameters, also to sort the venues by popularity and to choose only the *TopPicks* venues.

I appended the obtained venues as columns to create a new dataframe containing Latitude, Longitude and top trending venues.

For each house, then, I performed a new Forsquare call, retrieving this time the TopPicks venues present in a reduced radius from the house.

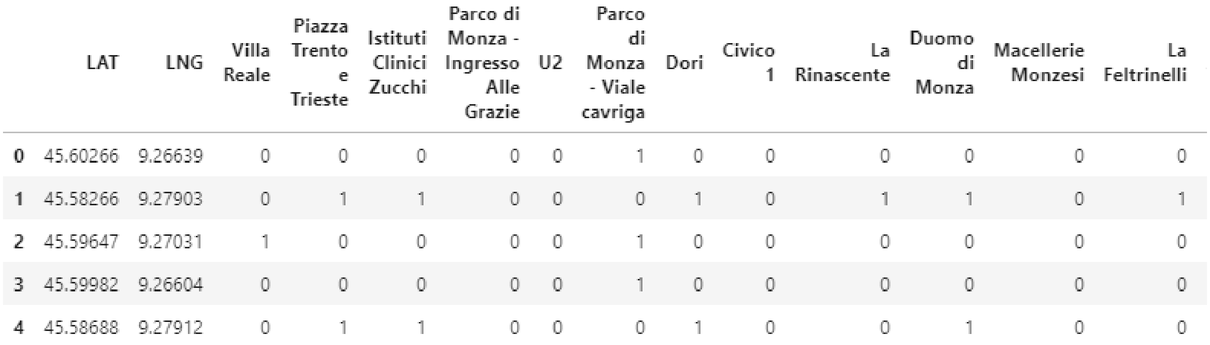
The obtained venues have been then compared with the top trending venues resulted from the initial call. Each column of the new dataset, corresponding to a specific venue, have been set to 1 if the venue was also present as venue of the Forsquare call of the current house.

Here is how the first 4 rows and columns of the new dataframe look like, before and after the Foursquare call for each house.



	LAT	LNG	Villa Reale	Piazza Trento e Trieste	Istituti Clinici Zucchi	Parco di Monza - Ingresso Alle Grazie	U2	Parco di Monza - Viale cavriga	Dori	Civico 1	La Rinascente	Duomo di Monza	Macellerie Monzesi	La Feltrinelli
0	45.60266	9.26639	0	0	0	0	0	0	0	0	0	0	0	0
1	45.58266	9.27903	0	0	0	0	0	0	0	0	0	0	0	0
2	45.59647	9.27031	0	0	0	0	0	0	0	0	0	0	0	0
3	45.59982	9.26604	0	0	0	0	0	0	0	0	0	0	0	0
4	45.58688	9.27912	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2 - venues dataframe before houses calls



	LAT	LNG	Villa Reale	Piazza Trento e Trieste	Istituti Clinici Zucchi	Parco di Monza - Ingresso Alle Grazie	U2	Parco di Monza - Viale cavriga	Dori	Civico 1	La Rinascente	Duomo di Monza	Macellerie Monzesi	La Feltrinelli
0	45.60266	9.26639	0	0	0	0	0	1	0	0	0	0	0	0
1	45.58266	9.27903	0	1	1	0	0	0	1	0	1	1	0	1
2	45.59647	9.27031	1	0	0	0	0	1	0	0	0	0	0	0
3	45.59982	9.26604	0	0	0	0	0	1	0	0	0	0	0	0
4	45.58688	9.27912	0	1	1	0	0	0	1	0	0	1	0	0

Figure 3 - venues dataframe after houses calls

Considering the last row in the figures, that corresponds to the fifth house, only the 2th, 3rd, 7th and 10th of the first 12 venues have been set to 1.

This means that in the immediate vicinity of the house it is possible to find only "Piazza Trento e Trieste", "Istituti Clinici Zucchi", "Dori" and "Duomo di Monza".

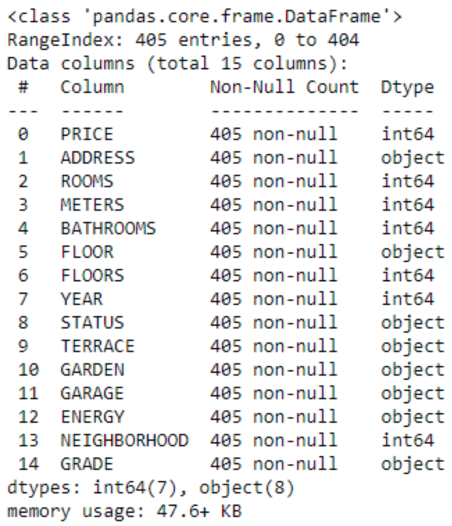
This dataframe has been used to cluster each house and as additional features to machine learning models as described in the following sections.

3. Methodology

3.1 Exploratory data analysis

The first step of the project has been to perform some exploratory data analysis.

By using pandas *info()* method I ascertained that no null values were present in the training set and all features were of the expected types.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 405 entries, 0 to 404
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   PRICE         405 non-null    int64
 1   ADDRESS       405 non-null    object
 2   ROOMS         405 non-null    int64
 3   METERS        405 non-null    int64
 4   BATHROOMS     405 non-null    int64
 5   FLOOR         405 non-null    object
 6   FLOORS        405 non-null    int64
 7   YEAR          405 non-null    int64
 8   STATUS        405 non-null    object
 9   TERRACE       405 non-null    object
 10  GARDEN        405 non-null    object
 11  GARAGE        405 non-null    object
 12  ENERGY        405 non-null    object
 13  NEIGHBORHOOD  405 non-null    int64
 14  GRADE         405 non-null    object
dtypes: int64(7), object(8)
memory usage: 47.6+ KB
```

Figure 4 - features info

The results confirmed that 405 non-null columns were present and that the features ADDRESS, FLOOR, STATUS, TERRACE, GARDEN, GARAGE, ENERGY and GRADE were categorical features and thus they will need some encoding.

To address the encoding, I used scikit learn OrdinalEncoder, with the following criteria:

ENCODED VALUE	FLOOR	STATUS	ENERGY	GRADE	TERRACE/ GARDEN/ GARAGE
0	GROUND	BAD	G	CHEAP	NO
1	MIDDLE	GOOD	F	NORMAL	YES
2	LAST	RENOVATED	E	EXPENSIVE	
3	VILLA	NEW	D		
4			C		
5			B		
6			A		
7			A1		
8			A2		
9			A3		
10			A4		

Table 1 - Features categorical encoding

All of these features contains values that range from low to high, thus it make sense to encode them using Ordinal Encoding.

For example, energy grade G is the category representing the highest consumption, A4 the category representing the lowest consumption. Therefore, we expect that houses of category A4 have higher prices then houses of category G.

Once performed categorical features encoding it has been possible to inspect their distributions by using histograms.

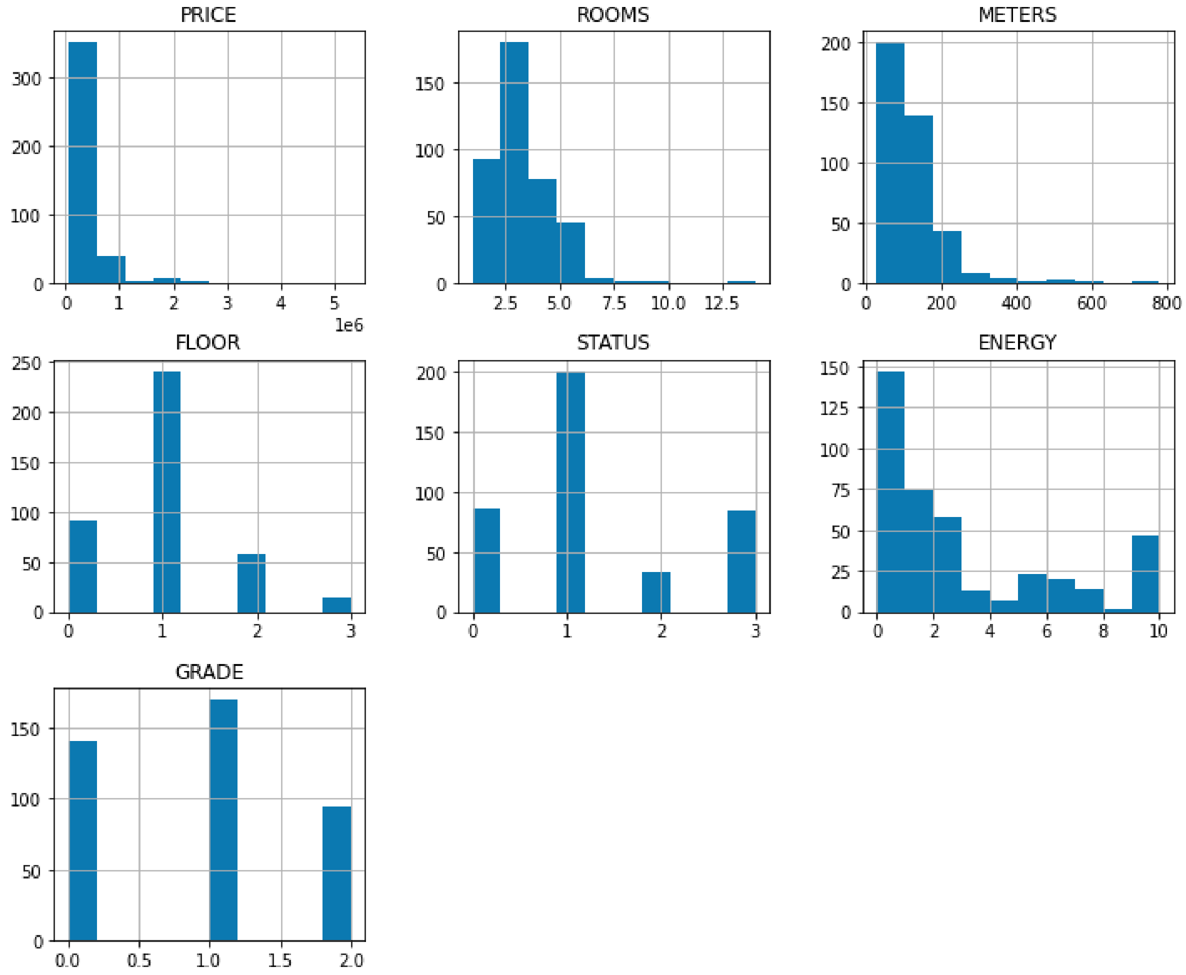


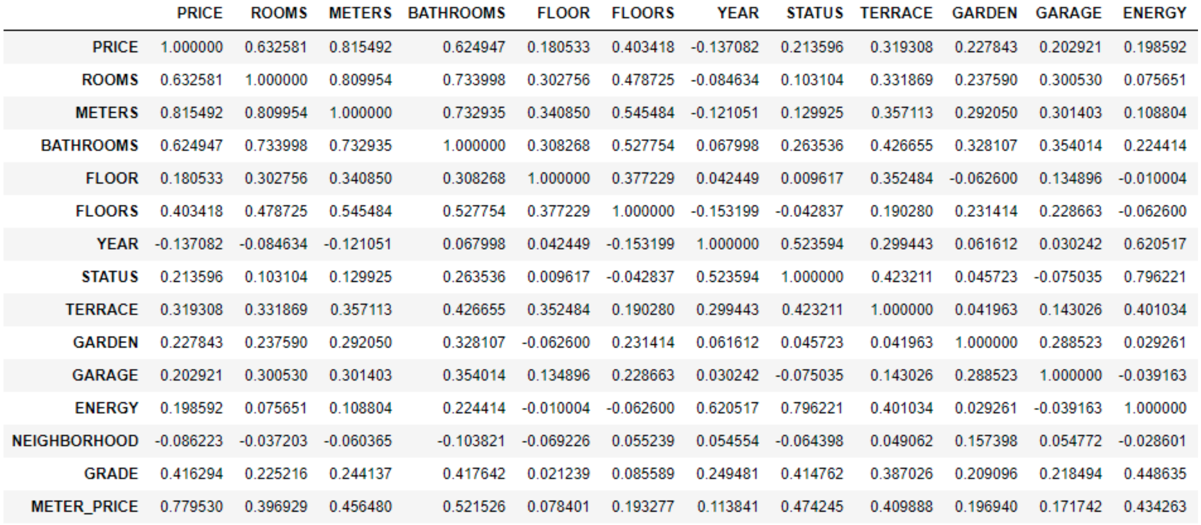
Figure 5 - Features distribution

After plotting the histograms distributions of PRICE, ROOMS, METERS, FLOOR, STATUS, ENERGY and GRADE, it can be seen that:

1. Most of houses prices are below 1000000 euros.
2. Most of houses have from 1 to 4 rooms
3. Most of houses have less than 200 square meters
4. Most of houses have 1 floor
5. Most of houses are in GOOD status
6. Most of houses have a bed energy class

With these distributions, the models will probably have higher accuracy for houses with these characteristics and they will probably make more mistakes if the house is particularly big or it is a new construction.

The second step has been to investigate if some correlation between features and target variable are particularly evident.



	PRICE	ROOMS	METERS	BATHROOMS	FLOOR	FLOORS	YEAR	STATUS	TERRACE	GARDEN	GARAGE	ENERGY
PRICE	1.000000	0.632581	0.815492	0.624947	0.180533	0.403418	-0.137082	0.213596	0.319308	0.227843	0.202921	0.198592
ROOMS	0.632581	1.000000	0.809954	0.733998	0.302756	0.478725	-0.084634	0.103104	0.331869	0.237590	0.300530	0.075651
METERS	0.815492	0.809954	1.000000	0.732935	0.340850	0.545484	-0.121051	0.129925	0.357113	0.292050	0.301403	0.108804
BATHROOMS	0.624947	0.733998	0.732935	1.000000	0.308268	0.527754	0.067998	0.263536	0.426655	0.328107	0.354014	0.224414
FLOOR	0.180533	0.302756	0.340850	0.308268	1.000000	0.377229	0.042449	0.009617	0.352484	-0.062600	0.134896	-0.010004
FLOORS	0.403418	0.478725	0.545484	0.527754	0.377229	1.000000	-0.153199	-0.042837	0.190280	0.231414	0.228663	-0.062600
YEAR	-0.137082	-0.084634	-0.121051	0.067998	0.042449	-0.153199	1.000000	0.523594	0.299443	0.061612	0.030242	0.620517
STATUS	0.213596	0.103104	0.129925	0.263536	0.009617	-0.042837	0.523594	1.000000	0.423211	0.045723	-0.075035	0.796221
TERRACE	0.319308	0.331869	0.357113	0.426655	0.352484	0.190280	0.299443	0.423211	1.000000	0.041963	0.143026	0.401034
GARDEN	0.227843	0.237590	0.292050	0.328107	-0.062600	0.231414	0.061612	0.045723	0.041963	1.000000	0.288523	0.029261
GARAGE	0.202921	0.300530	0.301403	0.354014	0.134896	0.228663	0.030242	-0.075035	0.143026	0.288523	1.000000	-0.039163
ENERGY	0.198592	0.075651	0.108804	0.224414	-0.010004	-0.062600	0.620517	0.796221	0.401034	0.029261	-0.039163	1.000000
NEIGHBORHOOD	-0.086223	-0.037203	-0.060365	-0.103821	-0.069226	0.055239	0.054554	-0.064398	0.049062	0.157398	0.054772	-0.028601
GRADE	0.416294	0.225216	0.244137	0.417642	0.021239	0.085589	0.249481	0.414762	0.387026	0.209096	0.218494	0.448635
METER_PRICE	0.779530	0.396929	0.456480	0.521526	0.078401	0.193277	0.113841	0.474245	0.409888	0.196940	0.171742	0.434263

Figure 6 - Features correlation

As expected, the price is mostly correlated with meters, since that that the price is computed by multiplying square meters with square meters price.

We then see that the square meter price has highest correlation values with BATHROOMS, STATUS, METERS, TERRACE, ENERGY, ROOMS.

Surprisingly the bathrooms number seems to have higher impact than expected on square meter price.

By plotting a scatterplot between rooms vs price and meters vs price, I found out that two outliers are present:

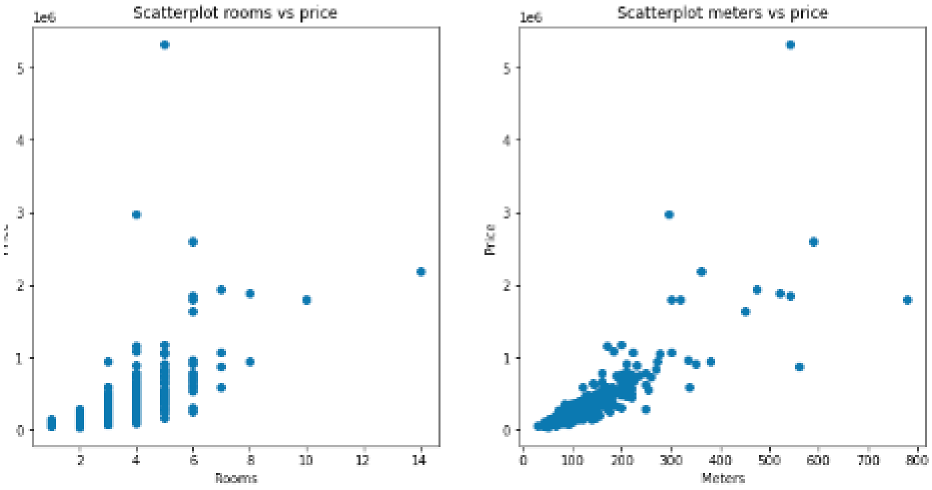


Figure 7 - Rooms vs price and meters vs price scatterplots

There is, in fact, one house with a price of more than 5 million euros, far above than the average prices of the houses with the same characteristics. There is also another house with a price of 3 million euros is quite far from the average price of the similar houses.

Their price is probably a mistake or it is due to very particular characteristics that are not captured by the selected features. For this reason they have been removed from the dataset.

As last step, after having retrieve latitudes and longitude of each house with geopy, by using folium I plotted all the training set houses in the map in order to explore spatial information.

I assigned them a different color depending on neighborhood and a different size proportional to the square meter price: the higher the price, the higher the CircleMarker radius.

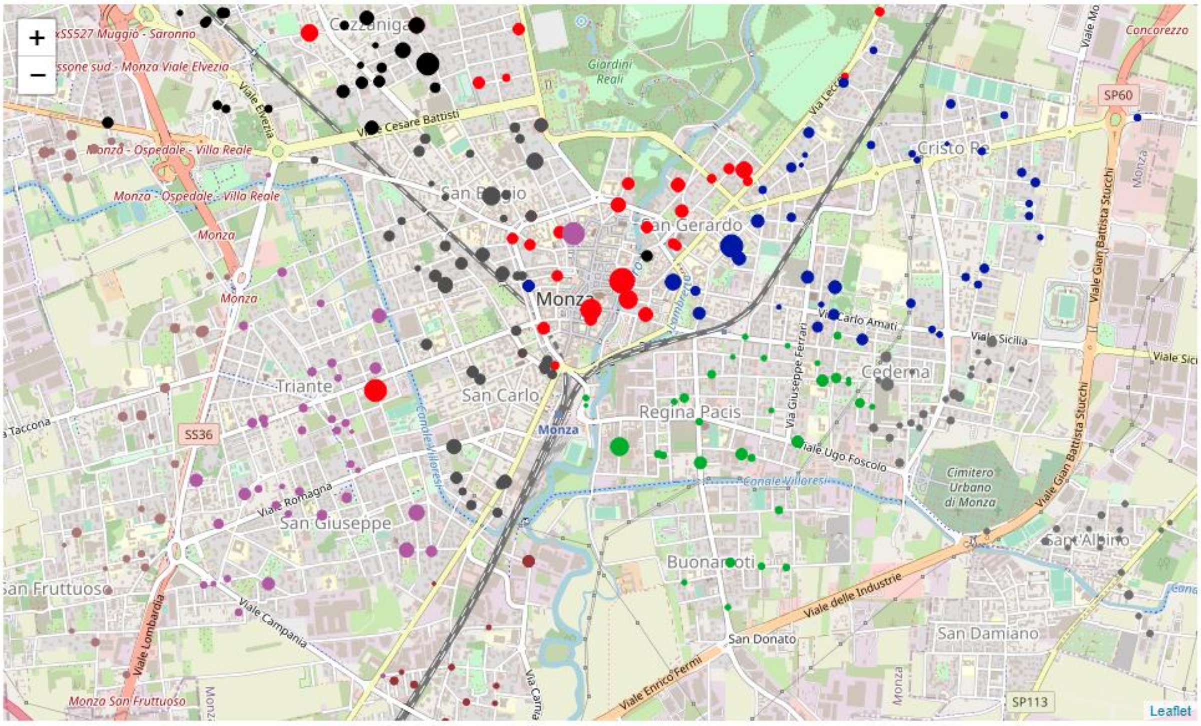


Figure 8 - Houses neighborhoods distribution

After plotting it can be seen that:

1. Houses have a good distribution around Monza territory
2. Houses with highest meter prices are in or around the historical center.
3. The neighborhood feature present in the dataset is not precise.
   In fact, there are different houses that have been mistakenly categorized: it is possible to see some "red" houses, that should correspond to the "center" neighborhood, that are quite far away from the center. The same happens with "blue" and "black" neighborhoods.

3.3 Cluster feature creation with K-Means on latitudes and longitudes

In order to use the neighborhood feature for the regression problem, after having ascertained that the actual feature present in the dataset was not precise, I decided to perform a K-Means clustering by using latitude and longitude coordinates.

I tested the results with the K-Means clustering ranging from 2 to 15 number of clusters.

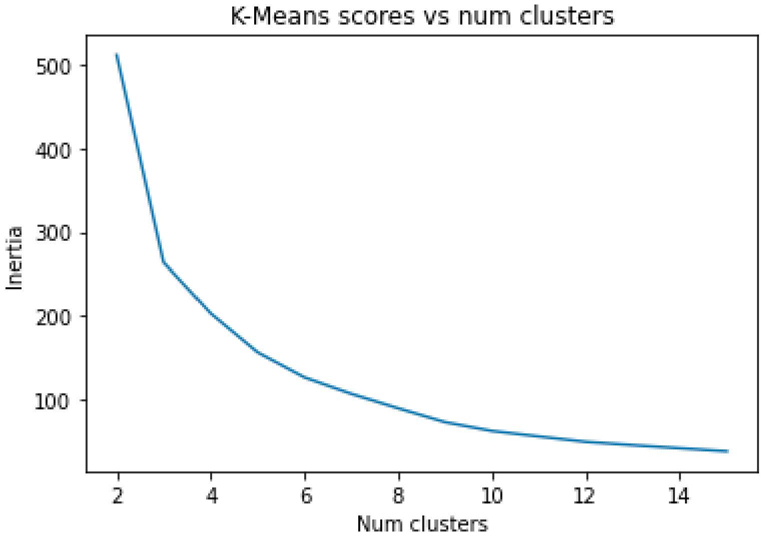


Figure 9 - K-Means clustering with coordinates

After having plotted on the map the results for all numbers of clusters, I determined that the best neighborhood division is obtained with 10 clusters.

This is the obtained map after K-Means clustering:

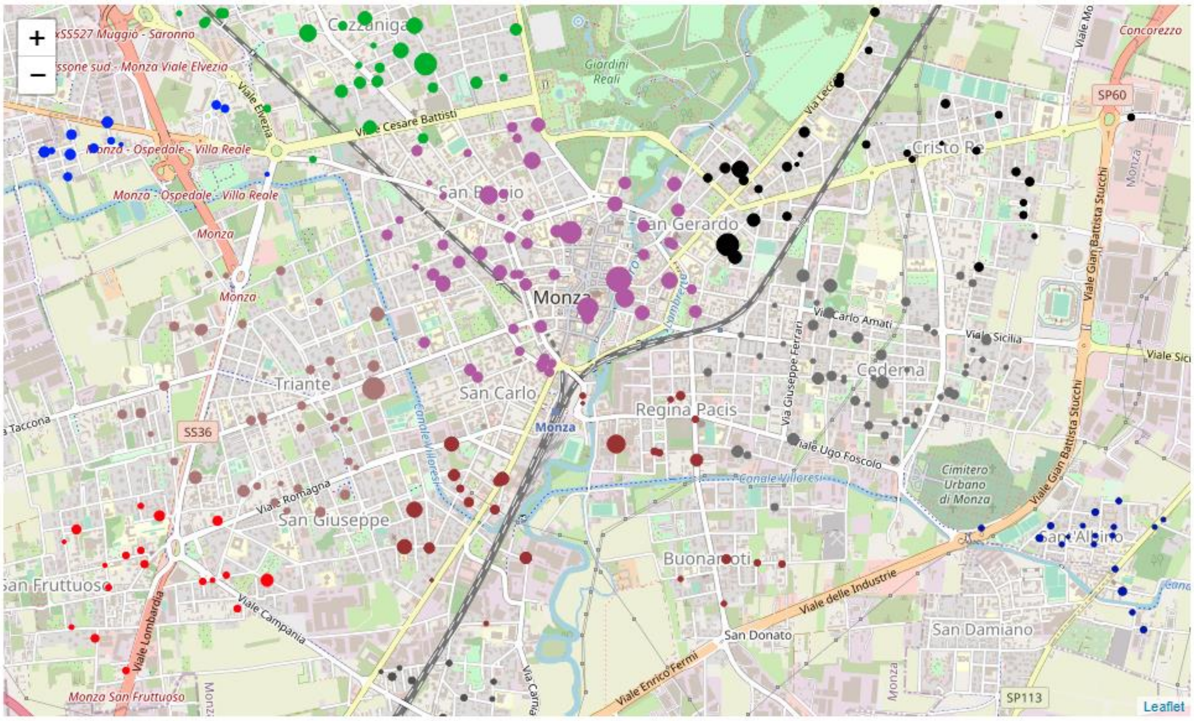


Figure 10 – K-Means clustering with coordinates houses division

Now the houses division better reflects the real neighborhoods division, so it can be used as a new feature for the regression analysis.

3.4 Cluster feature creation with K-Means with Foursquare trending venues

In addition to latitudes and longitudes, I wanted to perform also a K-Means clustering adding also the top trending venues around each house obtained with Foursquare API, with the process described in section 2.2.

As a recall, the idea was to retrieve the 30 top trend venues in Monza then, for each house, determine which of those are present in its immediate vicinity.

Therefore, 30 features will be added, one for each venue, and each house will have 1 or 0 to the i-th feature depending if the i-th venue is present or absent in the surroundings of the house.

The dataframe obtained is the one printed in Figure 3.

The performance, considering inertia, has a similar trend with respect to previous clustering, but with higher values.

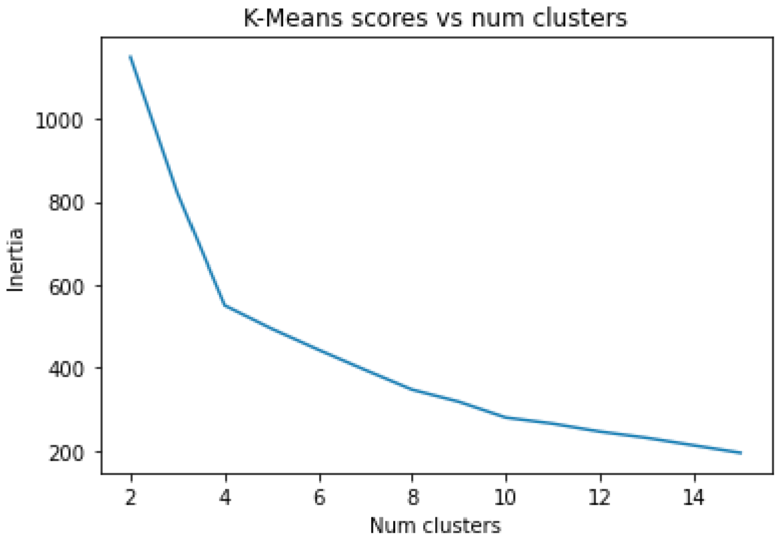


Figure 11 - K-Means clustering with top trending venues

After printing out the houses in the map after the clustering, I noticed that this time, the clustering has divided the houses in the center from the houses just around the center, obtaining smaller clusters in those areas.

Instead, the model lost the division of the peripheral neighborhoods obtaining big clusters.

This can probably be acceptable, since the peripheral neighborhoods have lower impact on square meter price.

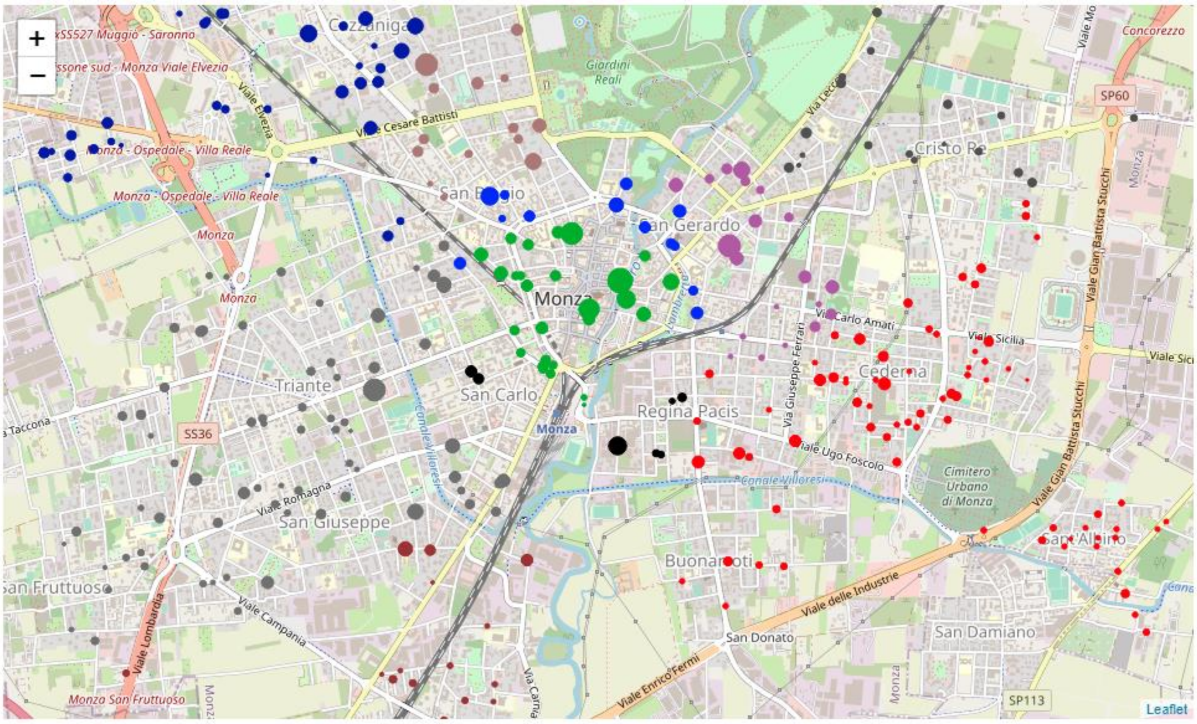


Figure 12 - K-Means clustering with trending venues houses division

3.5 Encoding FLOOR feature

Before proceed to the training phase of the regression models, as final step I wanted to perform a different encoding of the FLOOR feature.

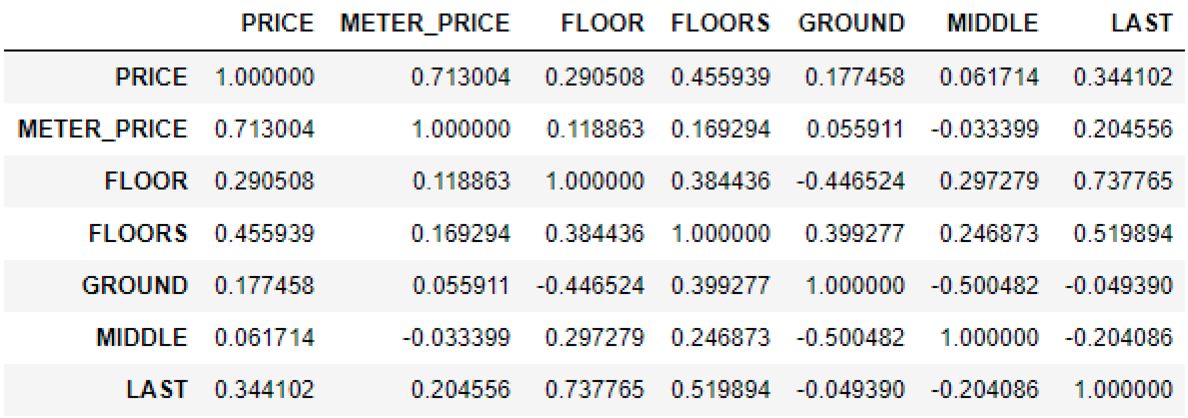
With the ordinal encoding, it seems that this feature is not really correlated to the square meter price, as seen in Figure 6.

This is something unexpected since, usually, houses at higher floors have also a higher cost with respect to houses at lowest floors.

For this reason, I decided to perform a different encoding creating three new columns namely GROUND, MIDDLE and LAST, that I populated basing on the values of FLOOR and FLOORS features.

In particular, if a house had a GROUND floor I set the GROUND column to 1, and the MIDDLE and LAST columns depending on how many FLOORS are present, and perform this reasoning also for MIDDLE and LAST. If there is a VILLA, instead, let's set all of the three columns to 1.

After this encoding, I computed again the correlation between these features and prices:



	PRICE	METER_PRICE	FLOOR	FLOORS	GROUND	MIDDLE	LAST
PRICE	1.000000	0.713004	0.290508	0.455939	0.177458	0.061714	0.344102
METER_PRICE	0.713004	1.000000	0.118863	0.169294	0.055911	-0.033399	0.204556
FLOOR	0.290508	0.118863	1.000000	0.384436	-0.446524	0.297279	0.737765
FLOORS	0.455939	0.169294	0.384436	1.000000	0.399277	0.246873	0.519894
GROUND	0.177458	0.055911	-0.446524	0.399277	1.000000	-0.500482	-0.049390
MIDDLE	0.061714	-0.033399	0.297279	0.246873	-0.500482	1.000000	-0.204086
LAST	0.344102	0.204556	0.737765	0.519894	-0.049390	-0.204086	1.000000

Figure 13 - Encoded floors features correlation

The highest correlated of the new features is LAST: even if with still a low value it is higher than using the FLOOR feature before encoding, so it will be used instead of the others.

3.6 Regression

3.6.1 Training sets selection

After performing features encoding, and cluster features creations, I prepared 4 training dataframes, to check if adding spatial information results also in a performance gain.

In particular, the 4 dataframes contain the following features:

Dataset 1

This dataset is made by the most correlated features, without any spatial information.

It is expected to be the less performing dataset.

Following are the selected features:

ROOMS	METERS	BATHROOMS	LAST	YEAR	STATUS	TERRACE	GARDEN	GARAGE	ENERGY

Dataset 2

The second dataset is made by concatenating the features of the first dataset and the cluster obtained with K-Means using houses latitudes and longitudes.

It is expected to have higher performance with respect to Dataset #1

Following are the selected features:

FEATURES OF DATASET 1	K-MEANS CLUSTER USING LAT AND LNG

Dataset 3

The third dataset is made by concatenating the features of the first dataset and the cluster obtained with K-Means using houses with Foursquare top trend venues.

It is expected to have higher performance with respect to Dataset #1, while we will check what the performance with respect to Dataset #2 will be.

Following are the selected features:

FEATURES OF DATASET 1	K-MEANS CLUSTER USING TOP TRENDING VENUES

Dataset 4

The fourth and last dataset is made by concatenating the features of the first dataset directly with Foursquare top trend venues features.

It is expected to have higher performance with respect to Dataset #1, while we will check what the performance with respect to Datasets #2 and #3 will be.

Following are the selected features:

FEATURES OF DATASET 1	TOP TRENDING VENUES FEATURES

3.6.1 Regression models

After having created the 4 training datasets, they have been standardized with sklearn StandardScaler.

The models used have been the following:

1. Multivariate Linear Regression
2. Ridge Regression with 3rd degree Polynomial Features
3. Support Vector Regression
4. Random Forest Regression

Except for Multivariate Linear Regression, I performed some hyperparameters tuning with sklearn GridSearchCV, training with 5 folds cross validation.

For each model I computed the R2 score and the root mean squared error.

4. Results

4.1 Training results

During training, by using GridSearchCV, I evaluated the models with the following parameters:

- Ridge Regression:
  - Values of alpha 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000

- Support Vector Regression:
    - Kernel functions: linear, poly, rbf, sigmoid
    - Values of C: 0.1, 1, 10, 100
- Random Forest Regression:
    - Number of estimators: 5, 10, 50, 100, 200

Following are the R2 and RMSE of the 4 models with the 4 training sets:

		R2 Score	RMSE
Multivariate Linear Regression	Dataset #1	0.4968	643.610
	Dataset #2	0.5029	639.662
	Dataset #3	0.5057	637.906
	Dataset #4	**0.6386**	**545.379**
Ridge Regression + Polynomial Features	Dataset #1	0.4464	675.064
	Dataset #2	0.4721	659.197
	Dataset #3	0.4689	661.225
	Dataset #4	**0.7745**	**430.790**
Support Vector Regression	Dataset #1	0.4703	660.295
	Dataset #2	0.4893	648.365
	Dataset #3	0.4832	652.202
	Dataset #4	**0.5951**	**577.315**
Random Forest Regression	Dataset #1	0.9178	260.069
	Dataset #2	0.9352	230.81
	Dataset #3	0.9307	238.76
	Dataset #4	**0.938**	**225.83**

Table 2 - Training set scores

It can be seen that, as expected, the Dataset #1 had the worst performance.

Datasets #2 and #3 had comparable performances, so the clustering performed with the venues features did not have a tangible impact on the models performances.

Instead, for every model, Dataset #4 had the best performances.

Therefore, adding directly the venues features, instead of using a cluster retrieved from them, allowed the models to better learn the underlying relationships.

Regarding the models, it can be seen the Random Forest Regression outperformed the scores of the other models, with a RMSE for Dataset #4 of only 225.83 euros.

Following the distribution plots of the price prediction of Random Forest model and a comparison of the best models.

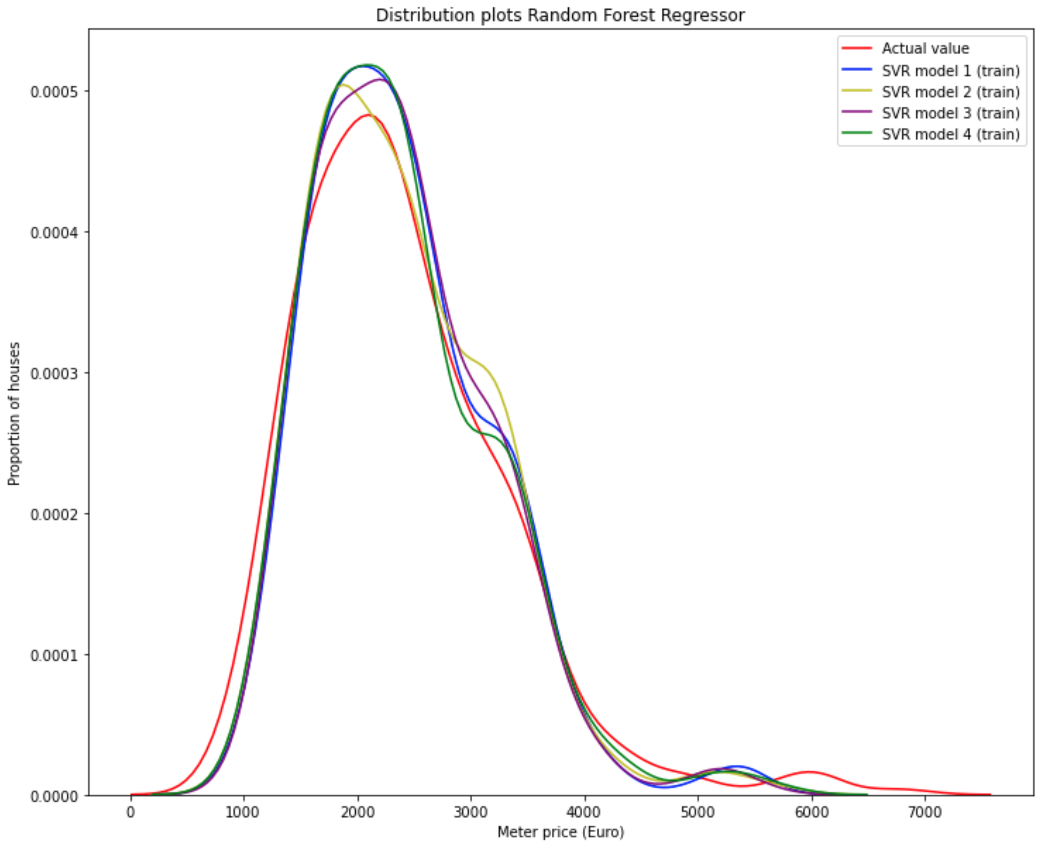


Figure 14 - Random forest predictions distribution plot

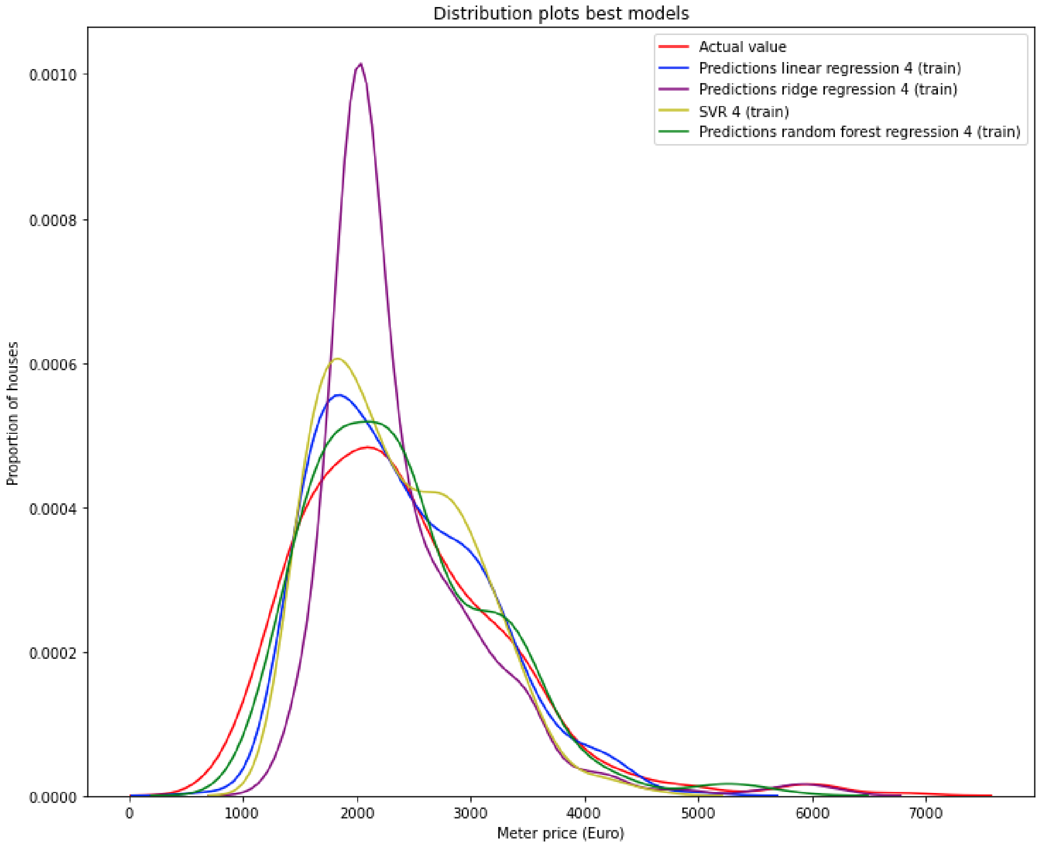


Figure 15 - Best models predictions distribution plots

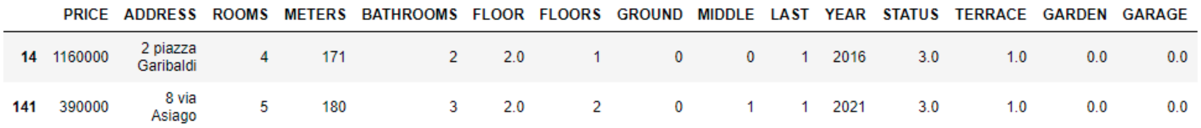
It can be seen that the training set distribution have an average square meter price of around 2000 euros.

It also have a non-linearity around square meter price of 6000 euros. While all the models failed to capture this characteristic, the ridge regression model came closer with respect to others. However, it assigned too many houses to the average price, with a greater error for cheaper houses.

The random forest model instead found the non-linearity around 5500 euros, but it has been the best model in estimating prices from 1000 to 4000 euros.

The other 2 models did not capture the non-linearity at all.

As last evaluation, I extracted the 2 houses with the worst price delta of the Random Forest model, that are the following:



	PRICE	ADDRESS	ROOMS	METERS	BATHROOMS	FLOOR	FLOORS	GROUND	MIDDLE	LAST	YEAR	STATUS	TERRACE	GARDEN	GARAGE
14	1160000	2 piazza Garibaldi	4	171	2	2.0	1	0	0	1	2016	3.0	1.0	0.0	0.0
141	390000	8 via Asiago	5	180	3	2.0	2	0	1	1	2021	3.0	1.0	0.0	0.0

Figure 16 - Worst predictions

They both are particular cases.

The first house, in fact, has a price of 1160000 euros but with only 4 rooms and 171 square meters, that results in a square meter price of 5758 euros. This house is quite expensive, as confirmed by the GRADE feature.

Vice versa, the second house is quite cheap, having a price of 390000 euros but with 5 rooms and 180 square meters, resulting in a square meter price of 2929.87 euros.

To have a visual inspection of the training errors, I created the delta price column, computes as the difference between the estimated price and the target price, and I plotted the training set with folium assigning to each CircleMarker a colormap from green to red depending on the delta price value.

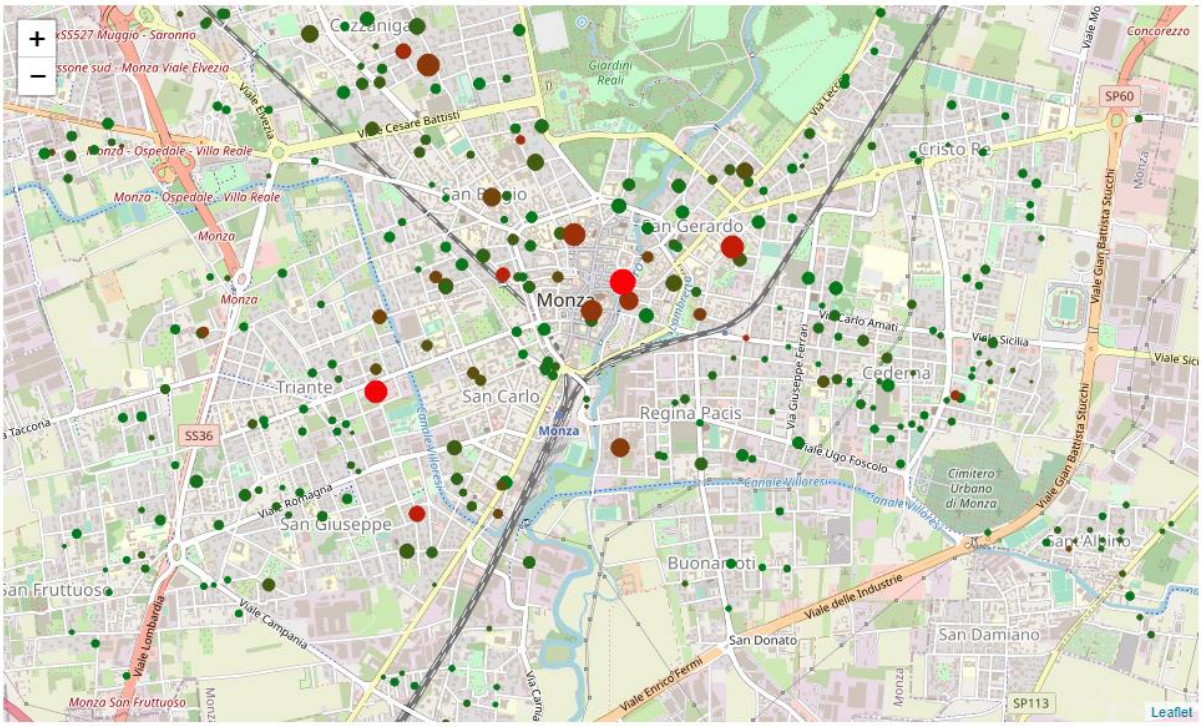


Figure 17 - Delta prices folium plot

It can be seen that the houses with the highest errors are also the houses with the higher square meter price, confirming that the model is not very accurate when the square meter price is above the average.

4.2 Test set evaluation

To test the generalization performance, the models have been evaluated on a test set of houses never seen during training phase.

The test houses dataset had the same format of the initial training dataset, so all the encoding pipeline and Foursquare venues features retrieval have been performed.

Following a folium plot of the test set houses:

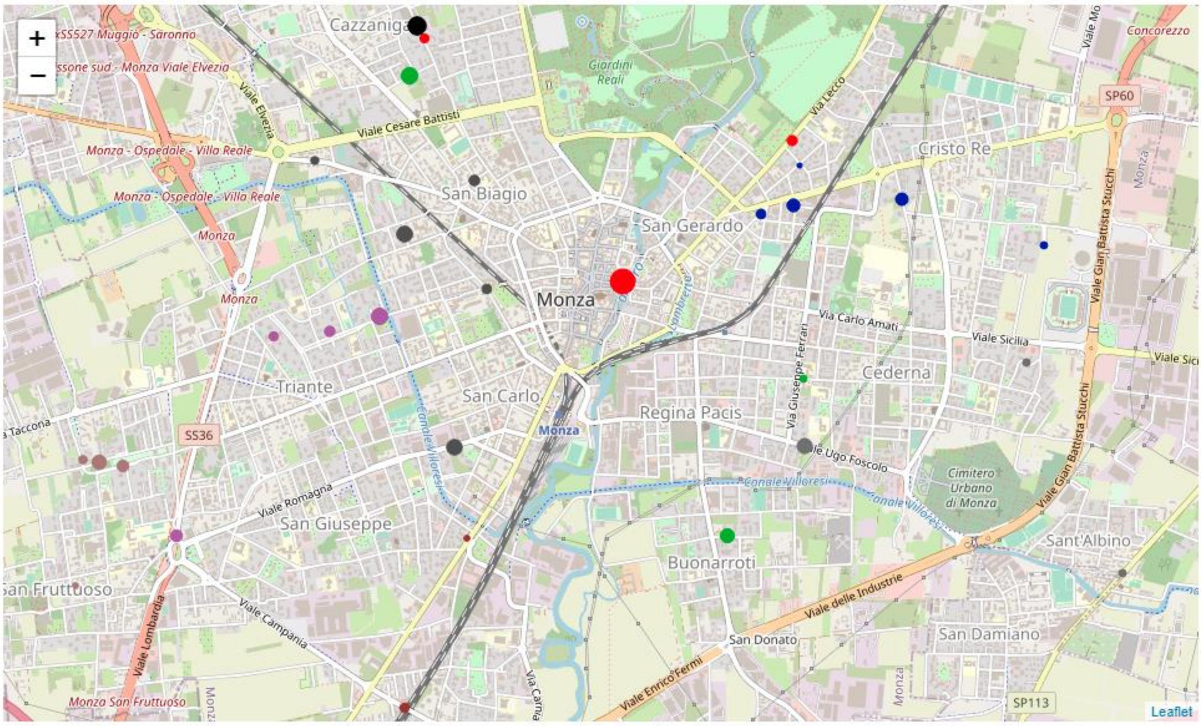


Figure 18 - Test set houses

It has been tested the performance only of the models with the best training scores, that are the models trained with dataset #4.

Following the test set evaluation results table:

		R2 Score	RMSE
Multivariate Linear Regression	Dataset #4	0.3129	752.609
Ridge Regression + Polynomial Features	Dataset #4	0.4639	664.784
Support Vector Regression	Dataset #4	0.4427	667.761
Random Forest Regression	**Dataset #4**	**0.6975**	**499.34**

Table 3 - Test set scores

The Random Forest model confirmed to be the best performing one, even if the score is quite far from the one obtained with the training set.

Following the distribution plots of the best models predicted prices with the test set:

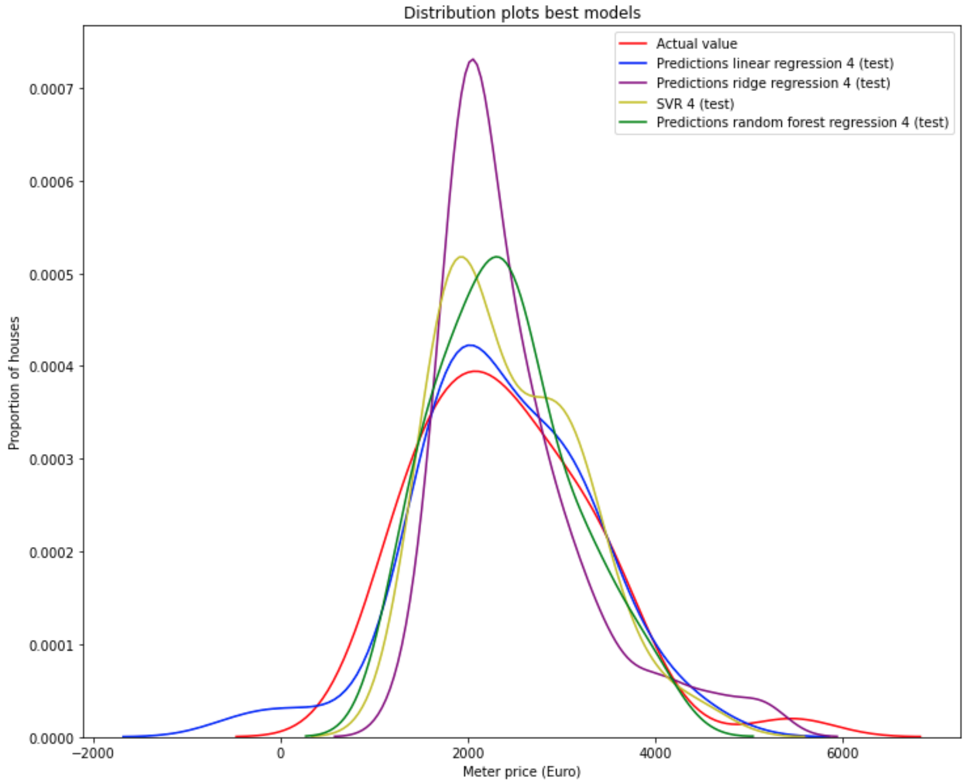


Figure 19 - Best models predictions on test set

5. Discussion

We have seen that even the best performing model fails to predict correctly high square meter prices. The main reason could be found in the lack of representative samples in that price range, therefore the training set can be largely improved in order to include more of those samples.

The usage of Foursquare allowed to capture meaningful spatial information that, when used as features, allowed the models to improve their predictive performance.

However, the main problem of using this approach is that the top trending venues change with time, so there should be found a way to stabilize those venues. In alternative, another way to get meaningful spatial information could be found.

6. Conclusion

In this project I evaluated the performance of 4 regression models with 4 different feature sets on the Monza house price prediction problem.

The results suggested that the usage of Foursquare retrieving a feature set indicating the presence of the top trend venues in the vicinity of the house can improve the models performance with respect of using only the house characteristics features. Finally, it has been found that Random Forest Regression model outperformed the other models scores.

However, there are many improvements that can be done, that can be summarized in the following points:

- The training set can be enlarged, by including more samples with high square meter price in order to let the model better capture the relations in that price range.
- The spatial information can also be improved. The retrieved top trend venues, in fact, can change over time. This can be a problem and it should be found a way to stabilize the top trend venues features.
- Other features of the house characteristics could also be added and the STATUS feature can be better stated.
- Other models can be tested and also the existing ones should be fine-tuned.
- It could also be possible to try a classification problem by predicting the GRADE feature using the house price as one of the features and not as target variable.