École nationale de la statistique
et de l'administration économique




Business data challenge report - group 4

# Prediction of a property price

*Students :*

Gregoire HUBERT

Nicolas JULIEN

Nick jofrein TEDONZE

Salah-Eddine EL MOUSLIH

Axel NAOARINE

April 2023

# Contents

# List of Figures

# 1 Introduction

## 1.1 Presentation of the project

### 1.1.1 Meilleurtaux

Meilleurtaux is a financial product brokerage company created in 1999 by Christophe Crémer. Initially present only online, Meilleurtaux widened its mode of distribution by creating a network of agencies from the 2000's. Currently, there are more than 300 of them all over France.

The company's objective is to advise individuals looking for financial services. Its goal is to put them in touch with the banking establishment offering them the best solutions in view of their profile and their project. Initially specialized in real estate loans, Meilleurtaux has over the years expanded its offer to other loans (e.g.: consumer), insurance, banking and savings.
In order to achieve this objective, Meilleurtaux offers a brokerage service, online comparison as well as a multitude of simulation tools directly accessible from its website to provide the as many information as possible to its customers.

Initially online only, Meilleurtaux keeps on with this strategy and seeks to be always more digital for an always more frictionless user experience. This is reflected through different axes:

- Explore data to give 360 degrees economic insights to customers and community.
- Build tools which bring intelligence to customers and staff.
- Build a data-ops ecosystem for all the group: datalake, shared AI and machine learning based tools, develops data science platform, API.

### 1.1.2 Project objective

It is in this context that this project takes place. The goal of the Business Data Challenge is to provide a tool for predicting the price of real estate according to its characteristics (type of property, location, surface area,...). This tool made available to customers of Meilleurtaux would allow them to estimate the real estate value of their dwelling. Thus, this estimate will be used in their profile in order to seek the best credits that could be proposed to them according to their project. This estimate is useful in two cases:

- In a mortgage loan project, when you need a bridging loan.
- In a credit consolidation project, when you have a mortgage guarantee.

### 1.1.3 Constraints & Challenges

In our opinion, this project presents 4 major difficulties to overcome:

- First, the subject contains a significant temporal component. Indeed, the price of a property can vary in time and the estimate of a property does not have a unique estimate depending on the time t when it is evaluated. Therefore, the prediction error actually contains two components :
  1) The hedonic prediction error, i.e. how far we are from the correct prediction without taking into account the temporal aspect, which means that we misunderstand how the characteristics of the property influence its price
  2) The prediction error of the price index, i.e. the extent to which the temporal variation of the price of the property has been wrongly predicted and therefore that the prediction of the global index of real estate prices at a given time is wrong.

  However, it is rather likely that Meilleurtaux is looking for a model that can minimize the first error, that is to say to determine the components that determine the price of a property rather than to have a model that just efficiently predicts real estate prices globally, which corresponds to two very different tasks. How to obtain a model that minimizes only the non-temporal estimation error ?

- This temporal dimension of the problem can also lead to a leakage problem when building our model. Indeed, when we manipulate a dataset, we have access to the whole price history. We have to be careful about the methodology used, especially in the construction of features, which may contain information about the future. Typically, it would be possible to obtain the price index for each year and thus obtain a very powerful model to predict what the price would have been at a past date. But the objective of Meilleurtaux is to obtain a model with good performance in inferability. In practice, it is only possible to obtain past data, and future data can only be predicted at best. How to avoid the leakage problem and be sure that the model remains effective when only present and past information are available ?

- In addition to the temporal dimension, the subject also contains a significant geographical dimension. It is certain that the geo-location of a property will make its price varyating, and that many geographic features can be constructed. Which ones to select and why? Also, the French territory is very heterogeneous concerning real estate and is actually made up of many sub-markets. The economic reality and the factors of variations will be very different for a property located in Nice and a property located in Paris. How to manage this heterogeneity and the differences between the real estate markets ?

- Finally, from a more practical point of view, the subject also presents an important stake of data comprehension and cleaning. Indeed, the data presented are very large and archaic, therefore, the appropriation of the information contained in the dataset, the feature selection, the management of missing values and outliers, the reinforcement mergers, are going to be very important steps with high added value for finding the best solution. There is no point in looking for the best model if the data has not been intelligently cleaned up beforehand. How can we best pre-process our data to get the maximum information out of it while limiting errors ?

### 1.1.4 Proposed Methodology

We chose an approach that aimes to address all of the above points as best as possible.

Thus, our first strong choice is to actualize the prices of our entire training dataset. By actualizing the prices, we ensure that each row of our dataset contains a price that is not impacted by the temporal dimension of the problem. Therefore, by training a model on that data, it minimizes the hedonic error mentioned above, which is the one that interests Meilleurtaux. The problem with this methodology is that it leads to data leakage: the discount is calculated using the variable of interest, the price of a property, and therefore a small part of the information carried by this variable is injected directly into the feature, which can bias the model.
It was therefore important to keep a part of the dataset not actualized in order to test the inference capacities of the model and to see how it performs when the prices are not updated, which is the case in real conditions. We then perform a "split" of the data by keeping the sales after the 2nd quarter 2021 undiscounted in order to obtain performance indicators closer to reality on our model.

For the geographical dimension, we decided to enrich the dataset by using different information to create new features containing information on the local environment of the properties. Indeed, in the original dataset, the rows contain only information directly related to the property, while in reality, the neighborhood can also have a strong impact on the price.
Finally, we decided to create different models that would be specifically trained on markets/types of property. Indeed, the economic realities that modulate the different markets can be too complex to understand for a single model, so we manually split up to create specialized models. It doesn't bother Meilleurtaux to have one model per city, and the only cost of this method is that it can add complexity to the code and increase computation time. So we limited ourselves to 20 models: one model for each metropolis in the 10 most represented in our dataset - Paris, Marseille, Lyon, Toulouse, Bordeaux, Lille, Montpellier, Nice, Nantes, Rennes; and for each major property type - house or apartement.

For the rest of the work, we adopt a "classical" approach of machine learning projects.

# 2 Literature review

The subject of estimating and predicting the price of a property is the source of many studies. Initially these studies were mainly based on econometric methods. However, in recent years, the use of machine learning methods has been included in an increasing number of studies. Moreover, we notice that this subject is universal. It can be found in a multitude of countries, each studying the real estate market in its own country but with similar results.

In the different studies, two types of data can be distinguished. The first is the data relating to the characteristics of the property. They can be : the surface, the number of rooms, the age of the property or in certain cases more precise data on the number of rooms according to their function (number of bathrooms, bedrooms, living rooms,...). These property-specific variables are effectively the most natural features to use to predict the price of a property. Nevertheless, the different studies agree on using variables outside the property that characterize the neighborhood. These variables vary from one study to another depending on the availability of data. They include:

- The social characteristics of the neighborhood
- Distance to public transport
- Accessibility to work
- The area in which the property is located (municipality, neighborhood,...)
- The population density in the area
- Type of street (to assess the view from the property)
- Quality of the school
- The facilities in the vicinity
- Level of crime
- Noise level

In the case of France, the studies aggregate these variables on the IRIS scale, which is the basis for the dissemination of sub-municipal statistics by INSEE. The authors choose to use the logarithm of the price per square meter as the dependent variable. Taking log makes the model more robust for prediction purposes, and the estimates become less vulnerable to extreme observations.

In the case of modeling, different methods are used depending on the purpose of the study. In the analysis of the influence of explanatory variables on the price of the property, the studies agree on using the hedonic regression model described by Rosen in 1974. Its operation is based on a vector of attributes, which may be empty or consist of a set of features, which is assigned to each characteristic or group of characteristics. Hedonic models can accommodate non-linearity, interactions between variables, or other complex evaluation situations. This is why they are regularly used in problems of estimating the effect of characteristics on real estate prices.

In the case where researchers are interested in predicting the price of the property, they favor the use of machine learning methods. In all of these, ensemble learning models, bagging or boosting get the best results. The random forest and Xgboost thus appear to be the preferred models for predicting the price of a property in view of the state of the art. These methods have the advantage of limiting overfitting, having good inferability and correctly capturing geographical effects. So, these are the same methods that will be studied in this project.

# 3 Data

## 3.1 Presentation and preparation of dvf database

### 3.1.1 The "demandes des valeurs foncières" database

The database "demandes de valeurs foncières" (dvf) gathers information on property values declared at the time of real estate transfers. The main information contained in the database is the value of the property, the address, the precise location (latitude, longitude), the *surface_reelle_bati* and the surface area of the land, the *nombre_pieces_principales* the type of properties (outbuilding, house, apartment and commercial or industrial properties. dfv database includes transactions from 2017 to june 2022. Some columns have a lot of missing data. These include the "lot" columns. For this category of columns it does not necessarily mean that data are actually missing but this comes from the definition of "lot". We will come back to their usefulness and their treatment in the part of preparation of the dvf table. Missing data are not a problem for most of the columns because they are not interesting (*nature_culture*, *ancien_nom_commune*, ...). Missing data are more of an issue for some other columns. This is notably the case for *surface_reelle_bati* and latitude-longitude variables. As we only miss about 1% of localisation variable and the fact that they are indispensable for our project, we choose to drop the corresponding observations. We will need to have a more precise look at *surface_reelle_bat* variable in the preparation step.

### 3.1.2 Preparation of dvf

The first step is to clean the dvf database. This cleaning follows several steps. After each step, we monitor the state of our database: quantity of deleted properties, and this metropole by metropole to make sure we keep enough data for the estimation.

Recovery of the most represented cities in the dvf database

We focus all our future work on estimating the prices of properties located in the most important zones. The zones are defined as the "Établissements publics de coopération intercommunales" (EPCI). We have retained the 10 EPCIs most represented in the dvf database: Bordeaux Métropole, Montpellier Méditerranée Métropole, Métropole Européenne de Lille, Métropole Nice Côte d'Azur, Métropole d'Aix-Marseille-Provence, Métropole de Lyon, Métropole du Grand Paris, Nantes Métropole, Rennes Métropole, Toulouse Métropole. These metropolises account for 18% of total transactions, before filtering on multiple properties and property types.
We choose to focus on a few zones after having discussed this point with Meilleurtaux and DataStorm. This choice is based on two points. First, it would be difficult to estimate the price of properties that are scattered across the territory. Most of the useful information for the estimation task are geographical ones and requires a minimum density of known transactions. This density

is mostly found in the biggest metropoles. Second, the goal of the project is to estimate the real estate price. Getting a good score in some important areas is more interesting for Meilleurtaux than struggling with low density areas.

Cleaning up multiple transactions

We only keep transactions involving houses and apartments. Other types of properties are not concerned by the Meilleurtaux project. To handle the DVF database it remains indispensable to clean, filter, and enhance our database with new features. Indeed , one of the first main goal is to deal with complex mutations. We have decided to work with mutations which include only one real estate based on the field "numero de disposition". To do that , we identify a unique mutation as the concatenation of *id_ mutation* and *date_ mutation*. We summarize our analysis with the following graph:

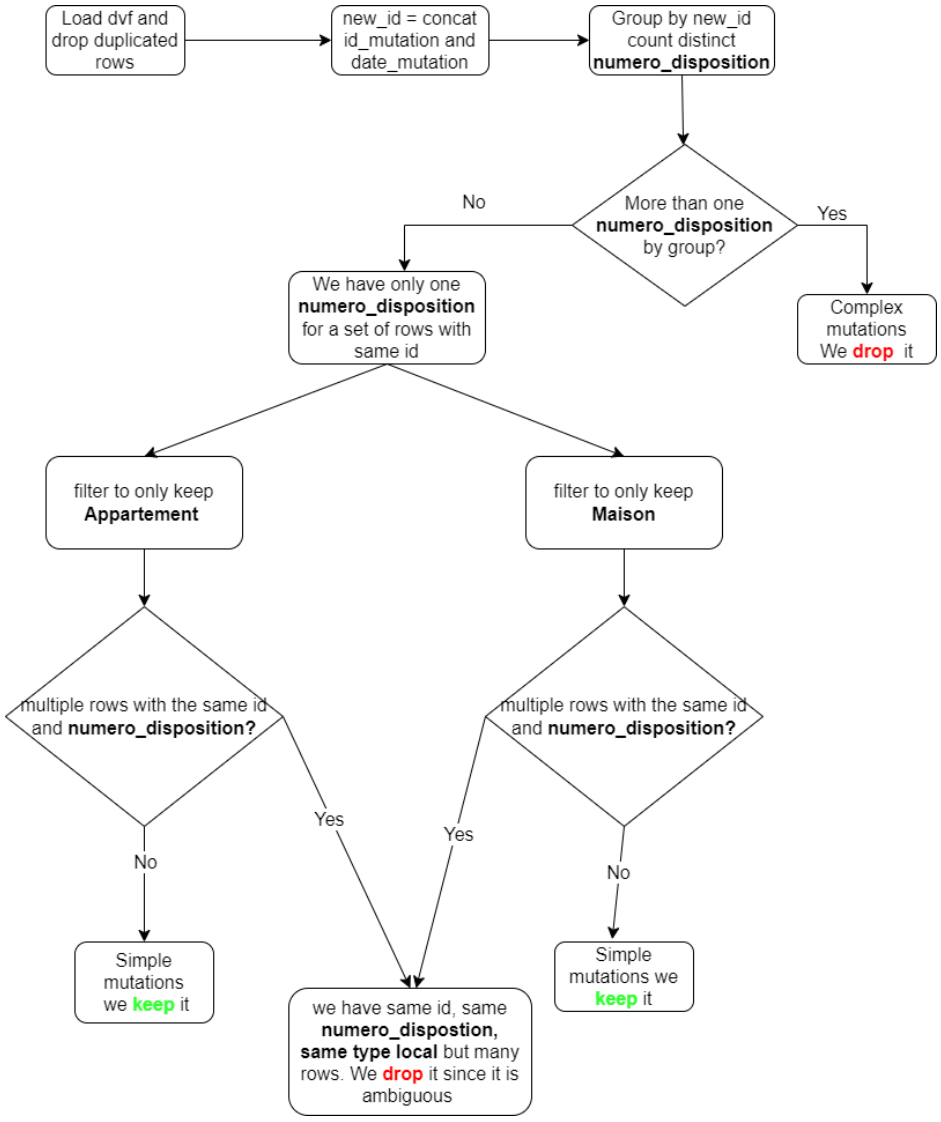
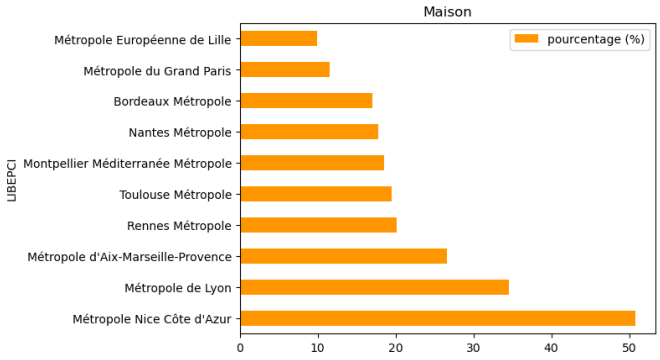
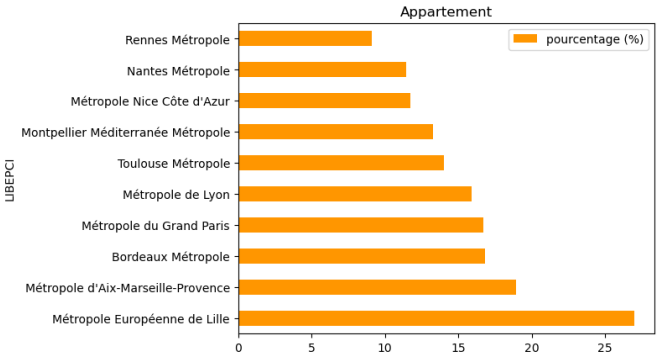



Figure 1: Summary of multiple transaction cleaning

We are left with 7% of the original dataset and we have droped 61% of the transactions of the previous dataset (after selction of metropolisis).
Once this transformation done, we can evaluate the rate of suppression in our dataset after having filtered our data on *nature_ mutation* sales and *type_ local* Appartement and Maison.




(a) Share of properties droped by filtering on houses




(b) Share of properties droped by filtering on apartments

Figure 2: Share of properties droped by filtering on houses and apartments

Price actualisation

As previously explained, temporality is an important aspect in this project. The working database gathers the sales of real estate in France since 2017. However, all characteristics being equal, it is not possible to compare a price per $m^2$ of a property sold in 2017 to a property sold in 2021. In fact, due to inflation and the evolution of the real estate market over time, the price scale can vary greatly. This is why the model will be trained on a database with prices corrected for this temporality. Thus, thanks to updated prices, all the properties in the database behave as if they had all been sold simultaneously. In this way, the model will predict the price of a property on the date of the update.

To update the prices, we use the real estate price index provided each quarter by the French National Institute of Statistics and Economic Studies (INSEE) in association with the Institute of Notaries. This one evaluates the evolution of the average real estate price through an index based on a base 100 in 2015. In order to make the update more precise, it is decided to update the prices according to the real estate zone in which the commune is located. Indeed, each French commune is classified in 5 real estate zones taking into account its attractiveness, its density, its population and other qualitative criteria. In addition, to improve the accuracy of the update on large cities (cities on which this project is focused) we decided to use the index provided for them. 4 index are available to us:

- Paris - apartments
- Agglomeration of Marseille - apartments

- Agglomeration of Lyon - apartments
- Agglomeration of Lille - houses

In order to make the update more precise, we decide to do it by type of property, house or apartment. However, for large cities, only one type of property index is provided. So, we use the national index for houses and apartments in order to estimate for these 4 cities, the type of property index not present in the initial database.

Finally, to update the price, we calculate the percentage change between the index of the quarter of sale and the index of the quarter of update. From this calculation, a price discount coefficient is applied to the price of the property value.

General filters on the dvf base

After having carried out these first cleanings aiming at keeping only the transactions concerned by the project (metropoles, house/apartment, multisales filter), we engage the filtering stage on the estimation variables and the target variable. The dvf database is subject to outliers, probably due to data entry errors. We therefore set up hard filters on the variables concerned: *surface_reelle_bati* and *nombre_pieces_principales*.
After studying the properties metropole by metropole and distinguishing the types of properties house/apartment, we choose different values for houses and apartments. The thresholds chosen match the 0.99 quantile for each properties type and metropole. We could have chosen thresholds specific to each metropole but they are quite homogeneous and the goal of this step is only to drop the inconsistent values in a general meaning, not a metropole specific one. Thus, for the rest of the project, we exclude outliers or exceptional properties of more than 360 $m^2$ or more than 10 main rooms for houses and of more than 200 $m^2$ or more than 6 main rooms for apartments. We are left with 6% of the original dataset and we have droped 18% of the transactions of the previous dataset (after filtering on multivente and *type_local*).

After having carried out the two steps of price update and split test train (step 3), we filter the properties on the basis of their price. Special attention must be paid to the problem of data leakage. Indeed, only train data must be used here to calculate the filter thresholds on these same train data. The price used for the train data is the discounted price as defined in section 3. For test data, we use the same quantile threshold as calculated on train data. This ensures that we do not use test data in our preprocessing steps. We are left with 6% of the original dataset and we have droped 3% of the transactions of the previous dataset (after filtering on *surface_reelle_bati* and *nombre_pieces_principales*).

## 3.2 Open databases used for the project and integration

### 3.2.1 Dataset Enrichissement : Iris Database and Equipments

In order to further enrich the dataset and include data concerning the neighbourhood of the property, information called IRIS is used. The IRIS code allows the French territory to be broken down into small homogeneous areas in terms of population or level of activity, at a finer scale than the simple commune code. The IRIS can be divided according to different criteria, there are 3 main ones according to the INSEE website:

- housing IRIS: their population is generally between 1,800 and 5,000 inhabitants. They are homogeneous in terms of the type of housing and their boundaries are based on the major breaks in the urban fabric (railways, waterways, etc.)
- Activity IRISs: they group together approximately 1,000 employees and have at least twice as many salaried jobs as the resident population
- miscellaneous IRIS - large specific areas with few inhabitants and a large surface area (leisure parks, port areas, forests, etc.).

Thus, the IRIS code corresponds to a kind of sub-municipality. It is then possible to merge the IRIS database with our dataset in order to give each dwelling an IRIS code using its geographical coordinates. For some IRIS the database is incomplete. We have chosen an analogue methodology to deal with these missing data. For all properties that miss an information we retrieve the information from its closest neighbor. This is a rigorous approach as the area we consider are quite dense and two close properties, despite not being in he same IRIS, will have very close values for the IRIS variables.

This IRIS code can then be used to link the dwelling to other databases such as the INSEE's permanent equipment base (bpe).

It is constructed from various administrative sources and lists a wide range of facilities and services, market or not, accessible to the public throughout France on 1 January of each year, in each IRIS. rq is what we do by year?

Thus, for each dwelling, we can construct features that list the number of facilities available in its IRIS. As the database contains 188 types of services, we decided to select the most relevant ones and to group them into large categories:

- Banks: Number of banks, savings banks, insurance companies in the IRIS...
- Post office

- Shops: Includes the categories "Hypermarket, Supermarket, DIY superstore, Supermarket, Grocery, Bakery, Butcher's, Deli, Frozen food, Fish shop" of the BPE.
- First Cycle Schools: Aggregates nursery and elementary schools
- Second Cycle Schools: Includes all middle and high schools
- General Practitioners
- Railway stations
- Cinemas
- Libraries
- Remarkable spaces and heritage

These categories seemed to be the most important for us in understanding the quality of the property. We preferred not to include all the facilities to avoid having too many features, which could cause overfitting.


Figure 3: Example of equipements in 18th arrondissement of Paris

### 3.2.2 Construction of a new variable: the average price per $m^2$ of the 10 nearest properties, for each property

After having carried out the two steps of price updating and train-split (step 3), we construct a new variable, the average price per $m^2$ of the 10 nearest properties, for each property. Each property in the dvf database is therefore assigned a new variable. The construction of this variable is motivated by the structure of the property markets. Prices per $m^2$ are largely determined by the location of the property. Thus, knowing the average price per $m^2$ in the area of the property whose price per $m^2$ is to be estimated should be a good basis.

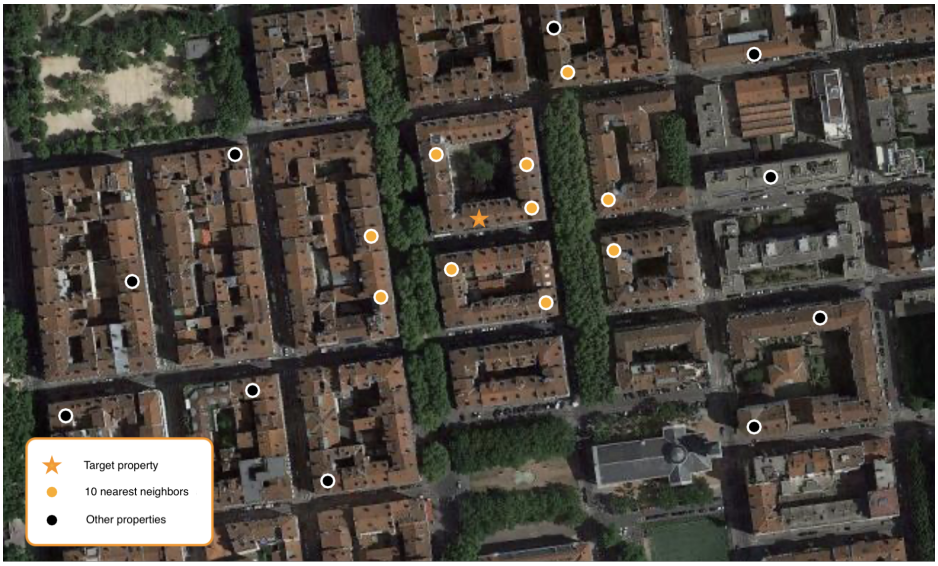



Figure 4: Example of the 10 nearest sold properties for the target property located at 1 Rue Bossuet, 69006 Lyon

Other criteria can influence the price per $m^2$ of a property, such as the *nombre_pieces_principales* and the total area. As properties that are close to each other may have different characteristics on these two variables, the price per $m^2$ of the area may be distorted. In order to eliminate the influence of these two criteria as much as possible, and thus to recover only the 'true' price per $m^2$ , we have set up a second method. This is also based on the 10 nearest properties. However, instead of simply calculating the average of the prices per $m^2$ , we run a regression of the price per $m^2$ on these two variables and on the data of the 10 closest properties. The intercept of this regression should represent the price per $m^2$ smoothed for the effects of the *nombre_pieces_principales* and the total area. It is therefore included as a new variable for each property. When predicting on final test data or in the actual use of the tool, the 10 nearest neighbors of the property whose price per $m^2$ is to be predicted will be searched in the historical dvf database.

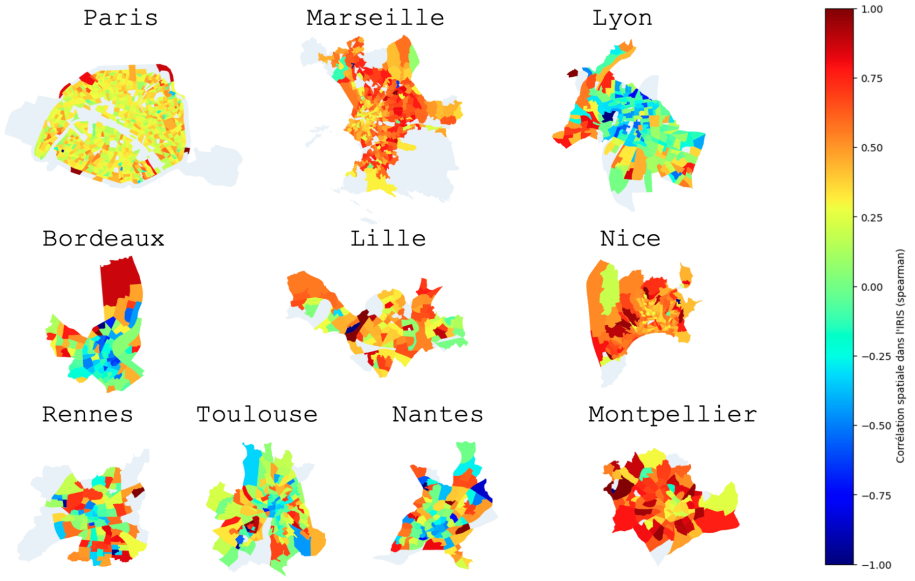



Figure 5: Correlation between *prix_m2* and *prix_m2_zone* in each IRIS

These graphs also show the spatial correlation in each city regarding prices per $m^2$ : in each IRIS, to what extent does the price of neighbours have an impact on my own price ?

Note: we decided to aggregate the $m^2$ prices of neighbours with this method in order to obtain a single indicator on the $m^2$ price of neighbours. We could also have included directly in feature *price_m2_nearest1*, *price_m2_nearest2*, etc. and let the model find itself the optimal calculation (be it an average, a median, a quantile, or an even more complex formula) to best combine the neighbours' prices. However, this represented too great a risk of overfitting, with variables that are highly correlated to our Y, but whose actual contained information is less clear and interpretable than a single index.

## 3.3 Stat Desc

### 3.3.1 Correlation of explanatory variables with target variable

We start by studying the correlation of each of our explanatory variables with our target variable, the discounted price per $m^2$ .

We can observe that the average price per $m^2$ in the area is the most correlated variable with our target variable. Next come the income variables in the IRIS (7th, 6th, 8th, 9th deciles) as well as the third quartile of income in the IRIS. The median income is also highly correlated. Some IRIS variables related to poverty are also highly negatively correlated: share of housing benefits in IRIS income in particular. The variables specific to each of the properties are not very correlated (surface area and number of rooms).

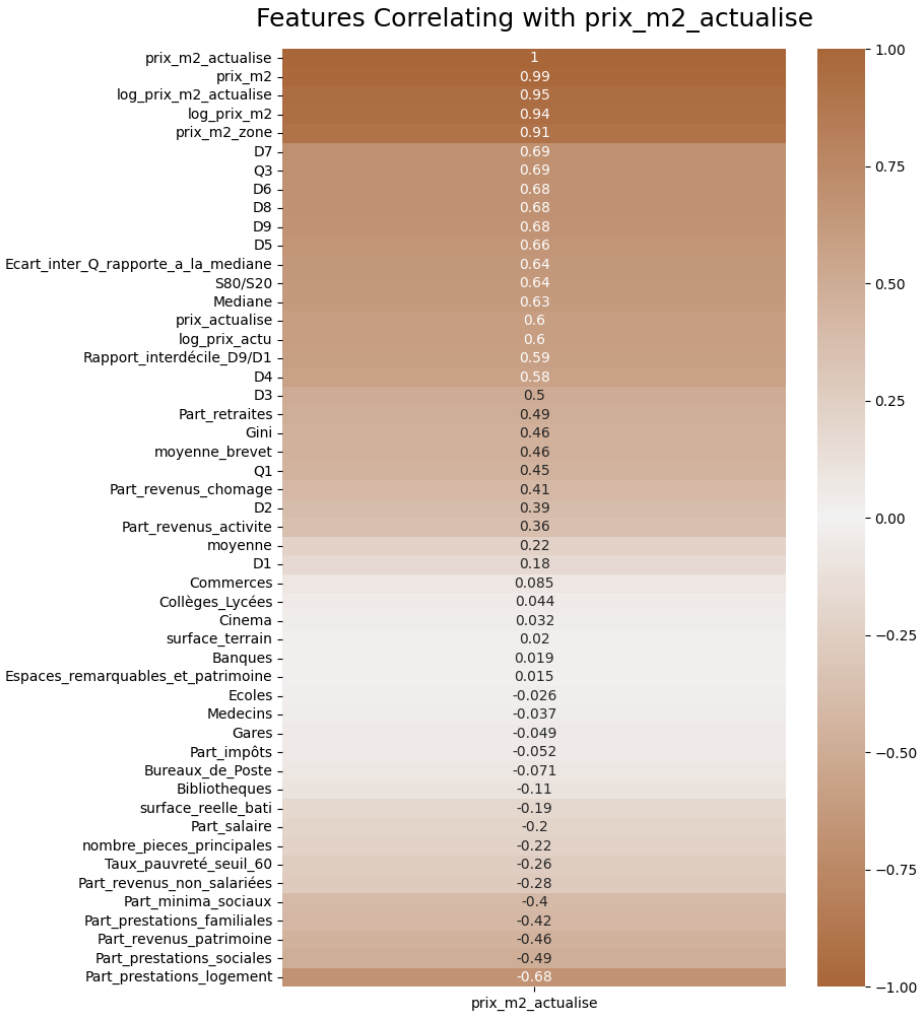



Figure 6: Correlation between each explanatory variable and the target variable (*prix_ m2_ actualise*)

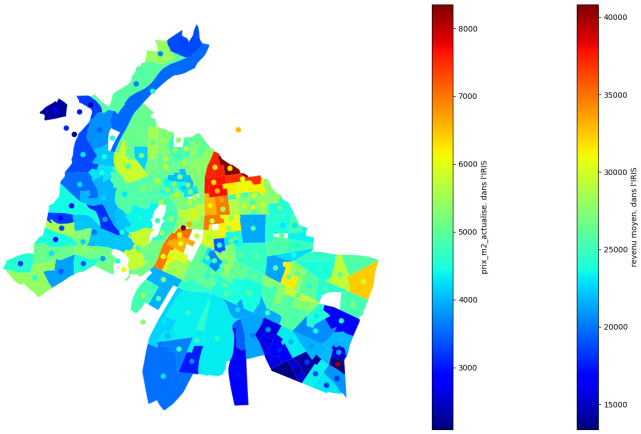
The levels of correlation and the order of the most correlated variables is different in each metropolis. This observation feeds our method of a model per metropolis and per house/apartment. For example, the correlations are much weaker for the metropolis of Rennes than for Paris. Moreover, the order is also very different for these two metropolises (see figure 19 & figure 20 in appendix). This observation is also true for the other metropolises. We will go into more detail in the study of these variables, starting with the environment variables.

### 3.3.2 Study of IRIS variables, the issue of metropolises and the link with price per m2

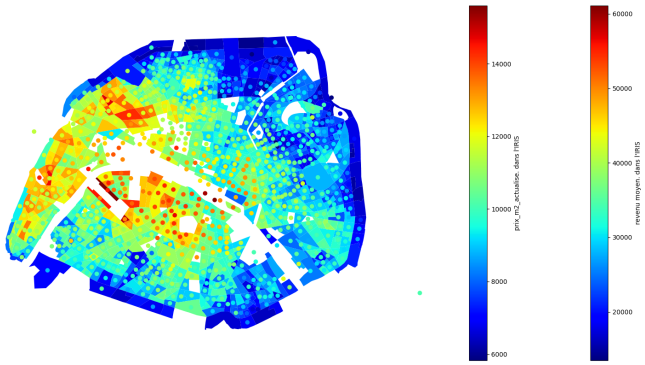
As we have seen, many of the IRIS variables are highly correlated with our target variable. Let us study them in more detail and explain why they are so. We take the example of four variables: the median income by IRIS, the inter-quartile range relative to the median, the seventh decile and the share of housing benefits.

Median revenu by IRIS and prix au $m^2$

We can observe on these two maps of Lyon and Paris (intra muros for reasons of legibility) that the median income of the IRIS (background color) does indeed seem to be positively correlated with the average price per $m^2$ of the properties in the IRIS (circles).




(a) Median revenu by iris and average price $m^2$ in each IRIS, Lyon



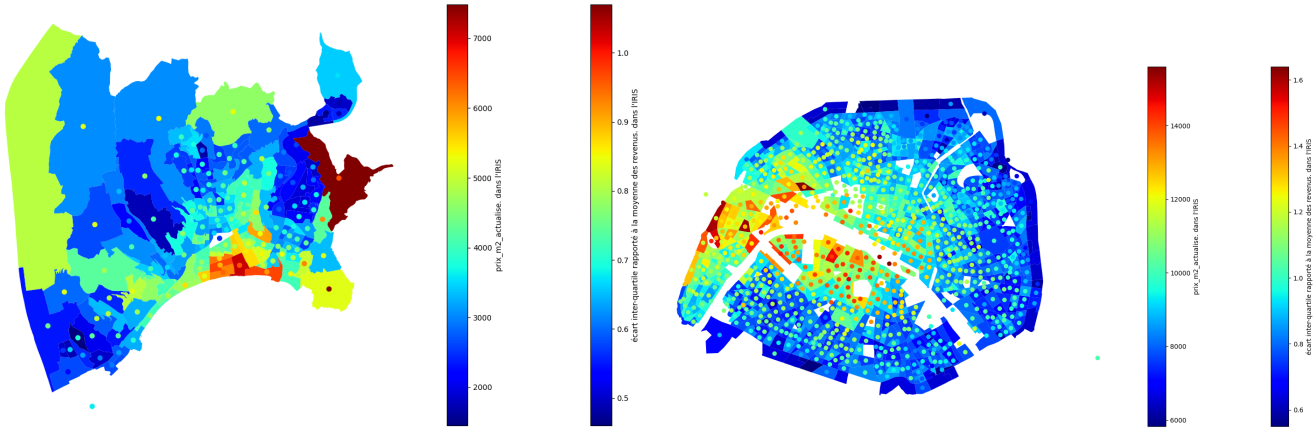

(b) Median revenu by iris and average price $m^2$ in each IRIS, Paris

Interquartile range to median and price per $m^2$

The interquartile range is a measure of income inequality within the IRIS. We can observe that the most unequal neighbourhoods are the neighbourhoods where the richest people live, notably the 16th arrondissement of Paris. The inequality here is accentuated by the fact that household employees often live in the same neighbourhood and therefore have a strong influence on local inequalities.

The strong correlation between the interquartile range and the price per $m^2$ must rather be due to relatively homogeneous neighbourhoods, where inequalities are rather low because all incomes are low on average. A low inequality is therefore very much linked to a low price per m2. This seems to be well verified in Paris.

However, in Nice (left map) strong inequalities are also linked to high prices, which is not really the case in Paris. There are therefore important differences between the metropolises. This point can be verified by looking more closely at the correlations between the variables and the price per $m^2$ by distinguishing between the metropolises. For the interquartile range, the correlation is much higher in Nice (0.41) than in Paris (0.27). These differences between cities justify the construction of different models for each city.



(a) Interquartile range to median and price per $m^2$, Nice

(b) Interquartile range to median and price per $m^2$, Paris

### 3.3.3 Study of types of properties (houses and apartments)

Houses and apartments are very different properties. We look at some statistics to see this.

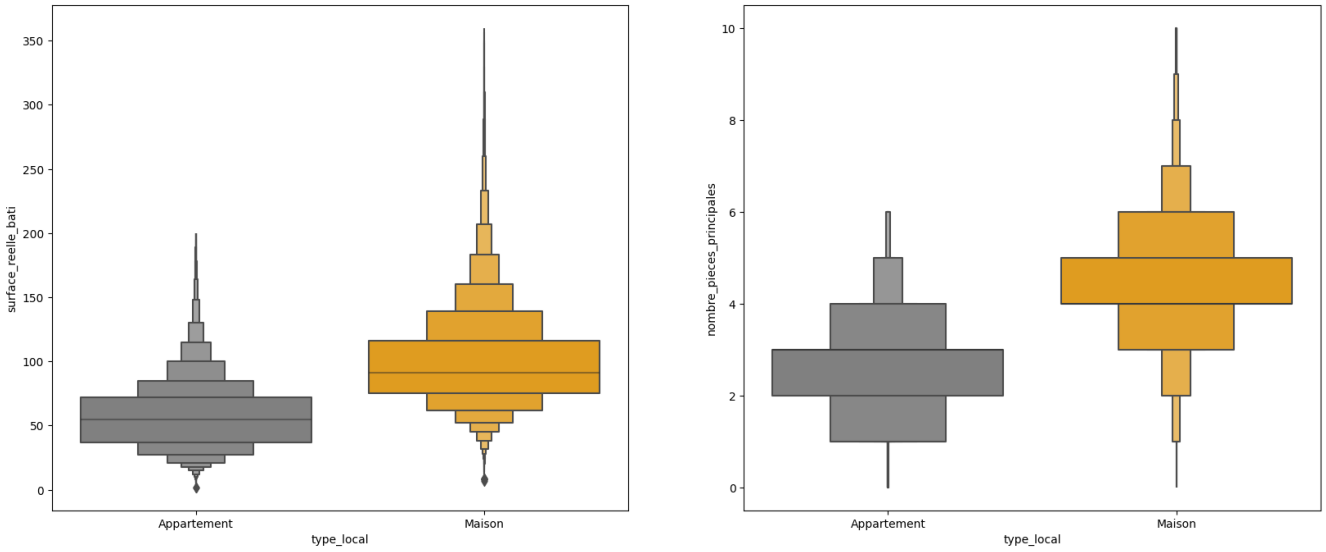



Figure 9: Distribution of surface and number of rooms by type of properties

The surface area of the property and the number of rooms (which are correlated at -0.19 and -0.22 respectively) differ greatly depending on the type of property. Separate models are therefore more appropriate to study these two different markets.

Next, we study the differences on these same variables between the metropolises (Figure 23 in appendix). We observe that the surface area for houses is relatively different. This is a new argument for building different models.

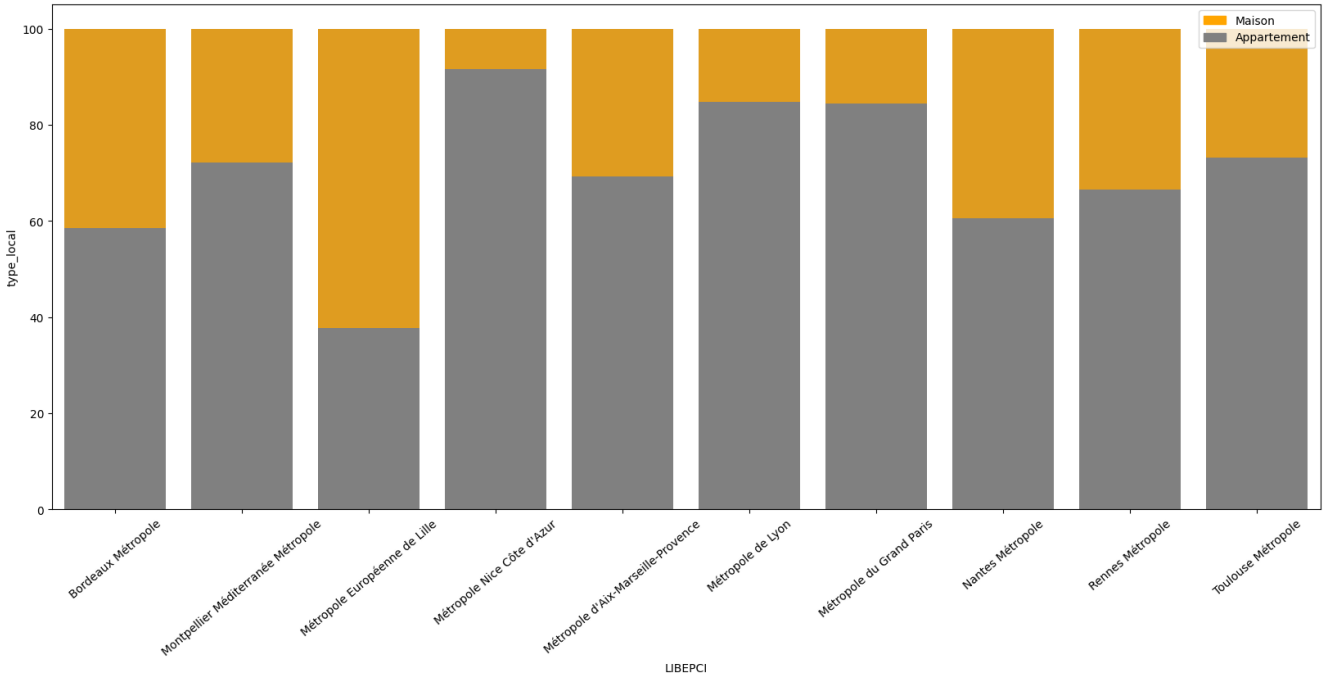



Figure 10: Share of each properties in all metropoles

Finally, the distribution of properties between houses and apartments is relatively unequal between the metropolises. Including both types of property in the same model for each metropolis would make it more difficult, if not impossible, to compare the results between the metropolises. Indeed, if our models estimate the prices of apartments well but less well those of houses, then the estimates for Lille will be less good in general than those for Nice, even though we may estimate the apartments of both cities equally well. This could lead us to disqualify the model for Lille whereas a Lille-apartment model could remain interesting.

### 3.3.4 Study of target variable

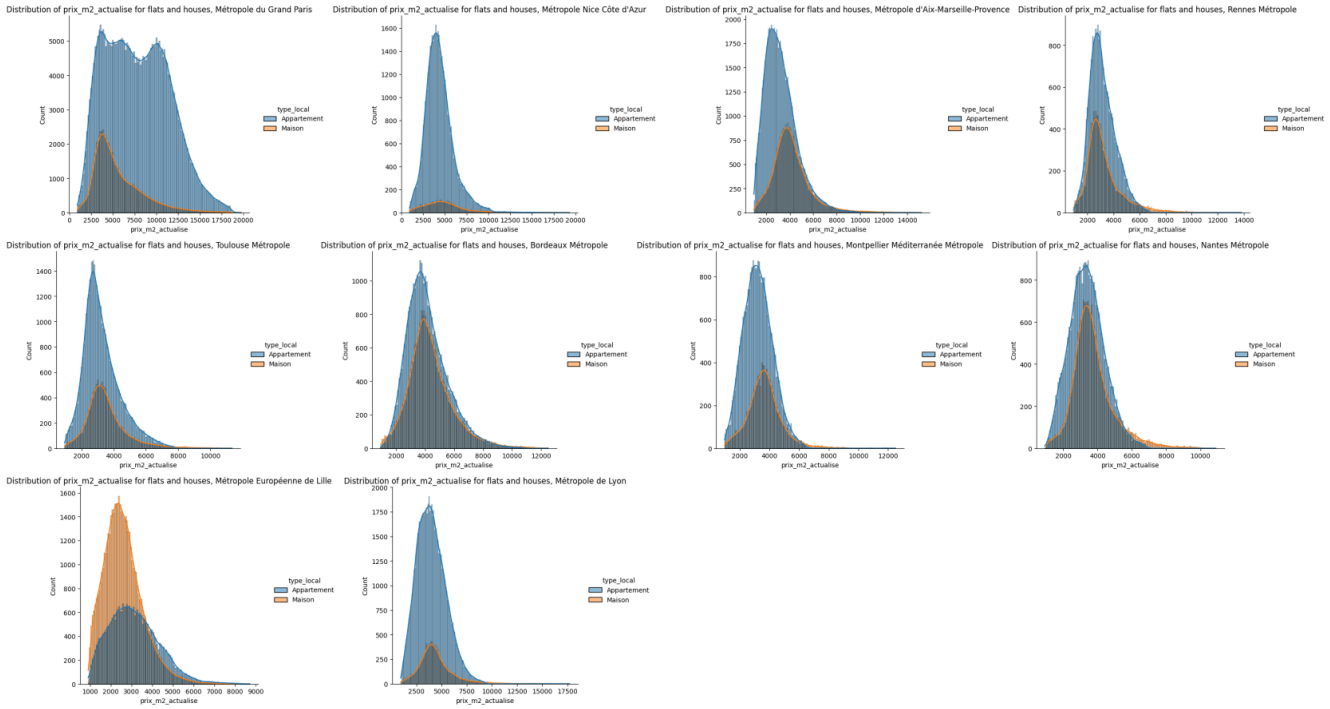



Figure 11: Distribution of price $m^2$ for each métropoles and house/apartments

We now look at the target variable, prix_m2_actualise. We see that the distribution is very different from one metropole to another and can also be very different between houses and apartments inside one metropole. This analysis supports our strategy to build one model for each couple metropole/house-apartment.

| Cleaning step | Share of transaction left vs original dataset | Share of transaction left vs previous step |
|---|---|---|
| Original dataset | 100% (18 036 812) | |
| Selection of metropolis | 18% (3 271 788) | 18% |
| Cleaning on multivente | 7% (1 269 091) | 39% |
| Filtering on surface and rooms | 6% (1 041 960) | 82% |
| Filtering on prices | 6% (1 007 243) | 97% |

Table 1: Evolution of transacion number through cleaning process

# 4 Modeling

## 4.1 Preprocessing

Before starting the modelling part of the project, it is necessary to make a selection of the data. Indeed, for the models described in the following section to work at their best, the raw data must be processed.

### 4.1.1 The working features

After the cleaning and enrichment carried out in the previous section, we select the most significant features present in the database capable of characterising the price of a property.

To predict the feature of interest, the price per $m^2$ of a property (*prix_m2*), the database is composed of 7 families of features :

- The price per $m^2$ of nearby neighbours (*prix_m2_zone*)
- Property characteristics features such as the actual living space of the property in square meters and Number of main rooms (*surface_reelle_bati, nombre_pièces_principales*)
- Nearby amenities features, the availability of these amenities can affect the value of a property (*Ecoles, Commerces, Gares, Banques, Medecins, Cinema,...*)
- Features of the quality of education of nearby institutions. This is the average of the middle schools for the brevet and the high schools for the Baccalauréat. (*moyenne, moyenne_brevet*)
- Geographical features, which define the location of the property. They can significantly influence the price of a property. ( *latitude, longitude, code_departement*)
- Income features of the IRIS. These features give an indication of the wealth of the IRIS, which as seen above impacts on the price of the property. (*mediane, D1, D9, Q3,...*)
- Social features of the IRIS. These features characterise the population living in the IRIS. (*Part_retraite, Taux_pauvreté_seuil_60, Part_minima_sociaux, part_prestation_sociales,...*)

### 4.1.2 Preprocessing

Pre-processing techniques are an essential stamp in data analysis and machine learning. They are used to prepare data for modeling by cleaning, transforming, and selecting relevant features. The following is an exhaustive list of common pre-processing techniques that we have used in our project:

- **Encoding**: converting categorical variables into numerical values so that it could be fitted to a machine learning model. For this we apply one-hot encoding to all the categorical features in our dataset. The only categorical variable used in the model is *code_ department* due to its low cardinality.Moreover, we build a model by metropolis and use the average price of the neighbour as a feature, So much of the location information is already included in this feature. Therefore it is not necessary to encode and use features like *commune* and *adresse_ voie* which have a high cardinality.

- **Feature Selection**: Used to select the most relevant features from a dataset to reduce dimensionality. We first calculate the correlation matrix, and remove highly correlated columns as well as unnecessary ones .Some features such as *longitude* and *latitude* are very correlated most of the time. In practice, for each metropolis and type of property, we assess the correlation between each variable and we remove one of the two features highly correlated between them (over 95%).

- **Standardization**: A technique used to transform the data so that it has zero mean and unit variance. This method helps to compare different variables on a common scale and thus improves the results of the model . In our case, we use 'StandardScaler'.
  The mathematical formula for StandardScaler can be expressed as:

$$z = \frac{x - u}{s}$$

  where $x$ is the input data, $u$ is the mean of the input data, $s$ is the standard deviation and $z$ is the transformed data after scaling.
  We have standardized all numerical columns in our model.

## 4.2 Split Train and test and Data leakage problem

Data leakage in machine learning occurs when information from the training data is inadvertently or intentionally included in the model, leading to biased or inaccurate results. There are several types of data leakage that can occur in machine learning:

- **Target leakage**: This occurs when information that would not be available during the prediction phase is used to train the model.
- **Data snooping**: This occurs when the model is trained using data that has been modified or filtered based on the test set. This can result in overly optimistic performance metrics.

To avoid leakage problem in our modelling we take some precautions:

- Divide the data according to their geographical position into 10 major metropolises.
- For each metropolis split Train and test according to time . In fact we find the best trimester to actualise our price so that we will have approximately 80% for the train set and 20% for the test set('2021-T2')(Table 2). So all data up to this trimester will be used as a test set and the price will not be actualised for this set.
- All pre-processing transformations (imputation of missing data, standardization, encoding, outlier detection) will be applied while maintaining independence between the training set and the test set. A good practice is to define all our transformations in a pipeline.

| Metropolis | Train_sample | Test_sample |
|---|---|---|
| Métropole du Grand Paris | 79,25% (335 975) | 20,75% (87 963) |
| Métropole Nice Côte d'Azur | 77,83% (44 634) | 22,17% (12 715) |
| Métropole d'Aix-Marseille-Provence | 87,78% (96 827) | 12,22% (13 485) |
| Rennes Métropole | 79,02% (27 866) | 20,98% (7 400) |
| Toulouse Métropole | 83,35% (48 788) | 16,65% (9 749) |
| Bordeaux Métropole | 77,19% (46 952) | 22,81% (13 874) |
| Montpellier Méditerranée Métropole | 80,67% (29 274) | 19,33% (7 014) |
| Nantes Métropole | 78,96% (41 766) | 21,04% (11 127) |
| Métropole Européenne de Lille | 78,49% (64 171) | 21,51% (17 586) |
| Métropole de Lyon | 81,33% (73 263) | 18,67% (16 814) |

Table 2: Train set / test set distribution

## 4.3 The models

We have decided to use two main families of models: Linear and ensemble learning models.

### 4.3.1 Linear Regression

Linear regression is a supervised learning algorithm used to establish a mathematical relationship between features and make predictions for continuous or numeric features. The primary objective

of this algorithm is to find the best-fit line that describes the relationship between the features.
One of the key benefits of linear regression is transparency and interpretability as it provides insight into the relationship between input and output features unlike black-box models that may produce accurate predictions but lack transparency.
The multiple linear regression model is the statistical tool most commonly used for the study of multidimensional data. As a particular case of a linear model, it represents the natural generalization of simple regression.

The multiple linear regression model can be mathematically represented as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon \tag{1}$$

Where $Y$ is the dependent feature, $X_1$, $X_2$, ..., $X_p$ are the independent features, $\beta_0$, $\beta_1$, $\beta_2$, ..., $\beta_p$ are the regression coefficients, and $\varepsilon$ is the error term.

In the context of retail, linear regression is used to predict house prices by analyzing the relationship between the price of a property and its various attributes. We decide to test this model as a first approach, as it is regularly used in the literature on the subject. Moreover, as explained, it offers a good interpretability of the results, which is important for MeilleurTaux.

### 4.3.2 The use of ensemble learning models to limit over-fitting

**What is an ensemble learning model?**

Ensemble learning models are based on a method that consists of combining a set of weak learners, learners that do not perform very well separately, often because of an over-fitting problem. That means, the generated algorithm is very sensitive to the training data, so the error obtained on this sample will be very low but it will result in poor quality predictions on a new data set because the model will be too specialized to the train set. This combination of weak learners allows to increase the learning performance of the model and to obtain a higher level of accuracy than the one that would be obtained by using separately one of these learners. Thus, these models have the advantage of limiting over-fitting by reducing the variance of the latter since the overall model is less sensitive to the training data. This is why the modeling of the project heads toward the use of ensemble learning models as recommended by the state of the art on this subject. Indeed, the objective of this project is to obtain the prediction of a property according to its characteristics. However, a property is rarely put up for sale several times over a short period of time. So, it is important to limit the over-fitting of our model in order to obtain the most accurate price predictions once the model is used on properties not present in our training database.

We could distinguish two main ensemble learning methods, the parallel and the sequential. Both will be tested in this project in order to select the one that gives the best results for the problem.

## Bagging

### What is bagging?

Bagging is a method that consists in combining a set of weak learners in parallel and then keeping the majority prediction within this set. To do this, bagging relies on the statistical method of Bootstrap. Bootstrap draws uniformly and with discount observations from the starting data set in order to create from it n different data sets. On each of these data sets, we randomly draw a number d of features on which the learner will be based. We then train a learner on each data set with x observations and d random features. Once the n different learners have been trained, a prediction is made simply by taking a majority vote.

### Random Forest and decision tree

The bagging method can be used on any type of predictive models to create weak learners. When it is used on decision trees, this model is known as Random Forest, which will be tested in this project. Thus, to predict the price per m2 of a property, the model will be based on a set of decision trees.

Regression using a decision tree consists in predicting the value of a new observation according to the leaf in which the algorithm will place this new property. To do this, the algorithm will construct decision rules based on the characteristics of the property in order to identify its price. The algorithm will proceed by iteration. During the first iteration, it will separate the observations into K groups (in this project, K=2) using the variable allowing the best split in order to explain the target variable, here the price per m2. This split will create K sub-populations corresponding to the first nodes of the tree. This splitting process will then be repeated several times on each group creating new nodes until the process stops when the maximum depth of the tree is reached or the regression cannot be improved. The result is the leaves of the tree corresponding to the average value of the price per m2 of the training observations in it.

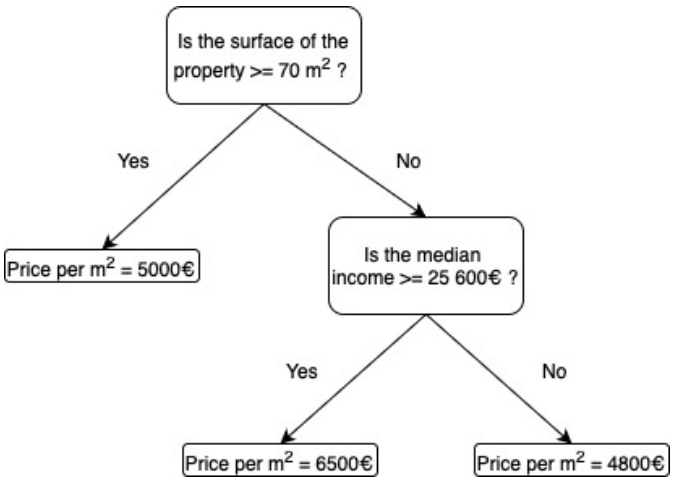



Figure 12: Example of a decision tree for a property

## Boosting

### What is boosting?

Boosting is a sequential set method, meaning that the weak learners used in boosting will be highly dependent on each other. Unlike bagging, in which the learners are trained independently of each other, in boosting, they are trained iteratively. Indeed, when the first regressor is trained, all observations have an equal weight. From the results of this first model, if an observation is badly predicted, its weight increases. Thus, thanks to this new weighted data set, a new learner is trained to try to correct the errors of the previous learner. This training and error weighting process continues until the entire train set is correctly predicted or the maximum number of models is added. The predictions of the last added model are then the global weighted predictions provided by the previous set of models.

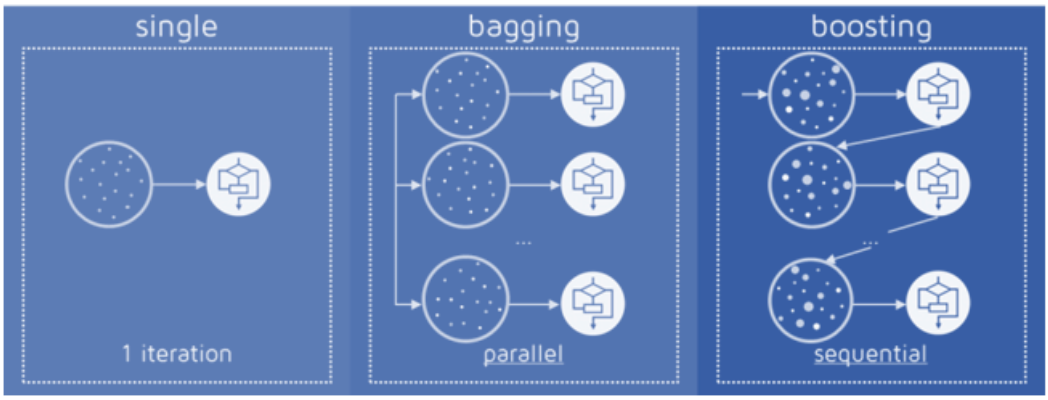



Figure 13: Differences in operation between ensemble learning models
source: *https://blent.ai/xgboost-tout-comprendre/*

**Gradient boosting**

Gradient boosting is a special case of boosting, in which the errors are minimized thanks to a gradient descent algorithm. Thus, only the first weak learner tries to predict the value of the observation. Indeed, the next models will try to predict the residuals of the observation from the previous model. That means, the difference between the predicted value and the real value of the observation. Moreover, Gradient boosting still respects the boosting principle of weighting the previously poorly predicted observations so that the new model concentrates its efforts on them. Finally, the final prediction will be the weighted sum of all the generated models. In the same way as the random forest, the weak learners used are decision trees, the principle of which has been explained previously.

## 4.4 Selection of the best parameters of models by cross-validation and selection of the best models

### 4.4.1 K-fold Cross Validation and scoring Metrics

K-fold Cross-validation is a technique used to estimate the accuracy of a predictive model in the cse of regression where we aim to make predictions.The main idea is to evaluate a model's performance on an independent dataset. Specifically, It works by partitioning the dataset into k-folds, training in k-1 folds and then testing on the remaining fold. We then repeat the process k times, with each fold used once for testing. Finally, we average the results of each iteration in order to obtain an overall estimate of model performance.

Cross-validation is practically useful because it allows us to assess the generalizability of a model. It also helps in selecting the optimal values for model parameters (hyperparameter tuning) to control its performance.

The mathematical formula for k-fold cross-validation is as follows:

$$\mathrm{CV}(n,k) = \frac{1}{k}\sum_{i=1}^{k} L_i$$

where $\mathrm{CV}(n,k)$ represents the cross-validation estimate of model performance using k-folds, and $L_i$ represents the loss (or error) of the model on the i-th fold.

In our case, we decide to use the R-squared metric to evaluate the performance of a model and to compare it. R-squared, or the coefficient of determination, is a common metric used in re-

gression analysis to evaluate the performance of regression models. The key idea of this technique is to measure the proportion of the variance in the dependent variable that is explained by the independent variables.

The mathematical formula for R-squared is as follows:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Where $SS_{res}$ represents the sum of squares of the residuals, which are the differences between the predicted and actual values of the dependent variable :

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

With $y_i$ is the actual value and $\hat{y}_i$ is the predicted value.
And $SS_{tot}$ represents the total sum of squares, which is the sum of the squared differences between the actual values and the mean of the dependent variable :

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

With $y_i$ is the actual value and $\bar{y}$ is the man value of the dependent variable.

In general, the higher the R-squared, the better the model fits the data. In the context of cross-validation, R-squared is a good metric because it measures the quality of the model fit to the data as well as a generalizability of the model which makes it a better model selection criterion for choosing between models.

### 4.4.2 Choice of the best model by type of property and metropole

For each metropole , type of property and type of model (xgboost, random forest , Linear regression model), we performed a cross validation with a Grid search to find the best combination of hyperparameters for our model. Then we choose from the 3 models the one with the best score on the cross-validation.

The choice of models is grouped in the tables below:

| Apartment | | |
|---|---|---|
| Metropolis | Best Model | Score $R^2$ |
| Paris | Xgboost | 0,80 |
| Nice | Xgboost | 0,50 |
| Marseille | Xgboost | 0,63 |
| Rennes | Xgboost | 0,64 |
| Toulouse | Xgboost | 0,65 |
| Bordeaux | Xgboost | 0,65 |
| Montpellier | Xgboost | 0,60 |
| Nantes | Xgboost | 0,60 |
| Lille | Xgboost | 0,72 |
| Lyon | Xgboost | 0,68 |

| House | | |
|---|---|---|
| Metropolis | Best Model | Score $R^2$ |
| Paris | Xgboost | 0,70 |
| Nice | Random Forest | 0,52 |
| Marseille | Xgboost | 0,49 |
| Rennes | Xgboost | 0,66 |
| Toulouse | Xgboost | 0,48 |
| Bordeaux | Xgboost | 0,49 |
| Montpellier | Xgboost | 0,36 |
| Nantes | Xgboost | 0,46 |
| Lille | Xgboost | 0,50 |
| Lyon | Xgboost | 0,48 |

It can be seen that for the majority of the models xgboost gets the best scores. Moreover, the scores are globally better for the apartment price prediction models.

# 5 Results

Now that we have trained models, it is important to evaluate them by analyzing their performance on the test data set, using different metrics to better understand the nature of their errors.

## 5.1 Evaluation

### 5.1.1 Metrics used

We decided to focus on 3 different metrics for model evaluation: the *Root-Mean-Squared-Error* (**RMSE**), the *Mean-Absolute-Percentage-Error* (**MAPE**) and the *Median-Absolute-Percentage-Error* (**MeAPE**). They were chosen to be evocative of the business impact that the models could have for maeilleurtaux.com

- **RMSE** : The formula of the *Root-Mean-Squared-Error* is:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{Y}_i - Y_i)^2} \tag{2}$$

We use this very classic Machine Learning metric because it is easily interpretable and penalizes outliers. This is particularly relevant in the context of our mission because meilleur taux.com seeks to avoid making large estimation errors, which can be costly to their client. Indeed, in the real estate sector, we can observe threshold effects on the amounts of the loans,

and it is more important to succeed in avoiding big estimation errors. The goal is to give an idea of the price of the property, but not to estimate it perfectly. By using the Euclidean distance, the RMSE will then penalize these large deviations more strongly, which is why we chose it.

- **MAPE** : The formula of the *Mean-Absolute-Percentage-Error* is:

$$\frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\widehat{Y}_i - Y_i}{Y_i}\right| \tag{3}$$

MAPE is also very relevant to our project. Where the RMSE gives an idea of the "raw" error of the estimate, the MAPE will allow us to obtain a metric calculated relative to the target variable. Indeed, it is not quite comparable to be wrong by 500€ on a price per square meter of 1000€ and to be wrong by 500€ on a price per square meter of 4000€. The MAPE allows to calculate the average relative error made by the model, which will have a lot of added value for the business. This metric allows for example to compare the performance of the model for all clients equally, where the RMSE favors good performance for clients with high priced properties. MAPE allows for a comparable metric for a customer with a small apartment and a customer with a large villa, which is important for Meilleurtaux in terms of fairness of treatment of their customers and inclusiveness. This is especially relevant since real estate loans are generally assigned based on price indices. Therefore, this error can provide a good estimate of the practical shortfall on a loan that a buyer could have obtained on the home in question. So this metric does have a strong business meaning.

- **MeAPE** : The MeAPE is simply the calculation of the median of the relative errors rather than the mean as seen previously. We likewise study the median because the MAPE suffers from some disadvantages related to its calculation method. First of all, the MAPE cannot be used when the values are close to 0, but this is rarely the case in our study. However, MAPE still always suffers from a problem : imbalance. The MAPE is indeed not symmetrical and is biased towards high values, since the target variable is present in the denominator. In fact, the MAPE can exceed 100% error when the error is greater than twice the price per square meter. Therefore, we are studying the median as an alternative, which allows us to penalize less the presence of outliers, which is natural while computing this error.

### 5.1.2 Performance assessment of the models on the test dataset

Here are the results of the models for each metric:

- **For apartments:**

Table 3: Metropoles' models evaluation on RMSE, MAPE and MeAPE, for apartments

| Metropole | RSME | MeAPE | MAPE |
|---|---|---|---|
| Métropole du Grand Paris | 1464.8 | 9.74% | 14.45% |
| Métropole d'Aix-Marseille-Provence | 647.0 | 11.02% | 15.37% |
| Métropole de Lyon | 756.3 | 11.55% | 14.32% |
| Bordeaux Métropole | 688.3 | **9.59%** | 13.12% |
| Métropole Européenne de Lille | 556.4 | 10.87% | 14.04% |
| Métropole Nice Côte d'Azur | 898.0 | 12.02% | **16.13%** |
| Rennes Métropole | 637.6 | **13.20%** | 15.17% |
| Toulouse Métropole | 560.0 | 10.24% | 13.31% |
| Nantes Métropole | 611.5 | 11.68% | 13.99% |
| Montpellier Méditerranée Métropole | 540.0 | 9.88% | **12.87%** |

- **For houses:**

Table 4: Metropoles' models evaluation on RMSE, MAPE and MeAPE, for houses

| Metropole | RSME | MeAPE | MAPE |
|---|---|---|---|
| Métropole du Grand Paris | 1266.9 | 12.51% | 16.05% |
| Métropole d'Aix-Marseille-Provence | 790.9 | **11.30%** | 14.95% |
| Métropole de Lyon | 992.9 | 13.40% | 16.05% |
| Bordeaux Métropole | 848.7 | 11.77% | 15.55% |
| Métropole Européenne de Lille | 562.7 | 14.14% | 17.67% |
| Métropole Nice Côte d'Azur | 1446.4 | **17.31%** | **24.48%** |
| Rennes Métropole | 648.2 | 12.14% | 14.64% |
| Toulouse Métropole | 677.2 | 11.35% | 15.14% |
| Nantes Métropole | 695.9 | 11.53% | **13.64%** |
| Montpellier Méditerranée Métropole | 722.2 | 12.44% | 15.26% |

Firstly, it must be acknowledged that our results are conclusive. The MeAPE ranges from 9.59% to 13.20% for apartments, and from 11.30% to 17.30% for houses. It is normal for the accuracy on houses to be slightly lower, as we have less data in our dataset. It is also normal for the MAPE to generally be higher than the MeAPE, because as we have mentioned, the nature of the formula used in the calculation pulls the errors upwards when the target variable is small.

We also note that despite the fact that the RMSE for Paris is quite high, the errors themselves are relatively small, as the price indices are just higher in Paris. Therefore, we can be satisfied that our model performs well in our largest cities (Paris, Marseille, Lyon...). We are even very successful for houses in Marseille, as well as apartments in Montpellier. Moreover our performance is satisfactory across all cities, but we still notice a small weakness in Nice, where the price per square meter is also generally high, but where we seem to capture certain factors less well than in

Paris. This is probably due to the coastal nature of the city, which involves factors such as distance to the beach or "sea view", which we did not take into account in the model. It would therefore be interesting to explore our model further to better understand the areas where it performs less well.

## 5.2 Interpretation

Now we can try to deep-dive our results to better understand our models and what are the drivers that lead them to make some mistakes.

### 5.2.1 Feature importance

Firstly, we can use the feature importance attribute of the model. Feature importance indicates how much each feature contributes to the model prediction. Essentially, it determines the degree of usefulness of a specific variable for the current model and prediction. It is calculated by analyzing the weak estimators of the XgBoost model (often decision trees) and looking at the entropy gain achieved by a split on this variable and the number of splits specific to this variable.

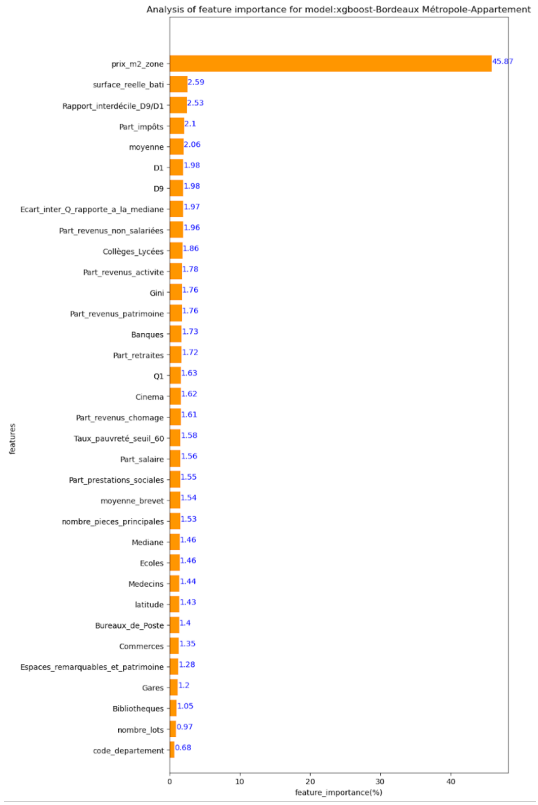



Figure 14: Feature Importance of the Bordeaux' apartments model

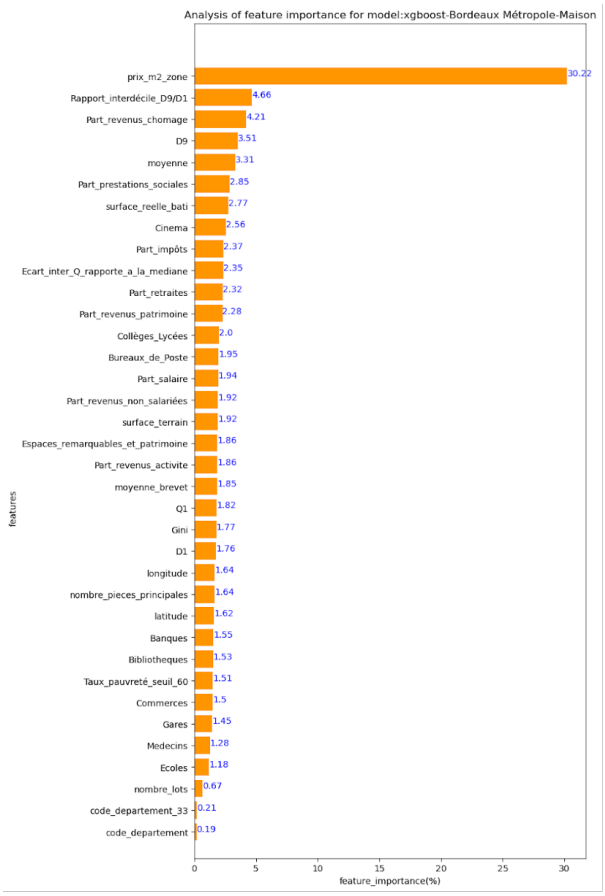



Figure 15: Feature Importance of the Bordeaux' houses model

We decided to take a closer look at the two models for Bordeaux, a city where we perform well. We notice, not surprisingly, that the price per square meter of the 10 closest neighbors has a significant impact on each model. We also observe this trend in the feature importance of all cities: the price per square meter of the area has very high importance every time. As for the other variables, they are generally slightly more important for houses than for apartments, indicating that the price per square meter of houses is less explained by that of the neighbors than for apartments.

Socio-economic indicators of the area often rank high, while amenities seem to have relatively low importance in all models. However, there is no clear hierarchy among the other variables, whose importance often varies depending on the city and type of housing.

### 5.2.2 Understanding of errors

Therefore, to truly understand what characterizes each model, we can try to focus on the nature of the errors made by each one.

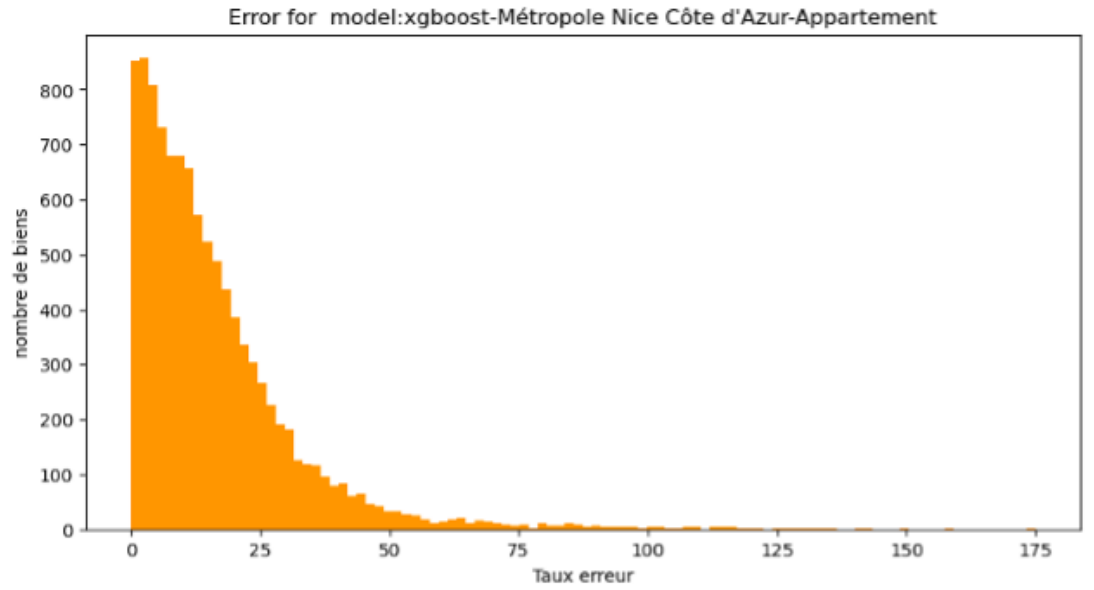
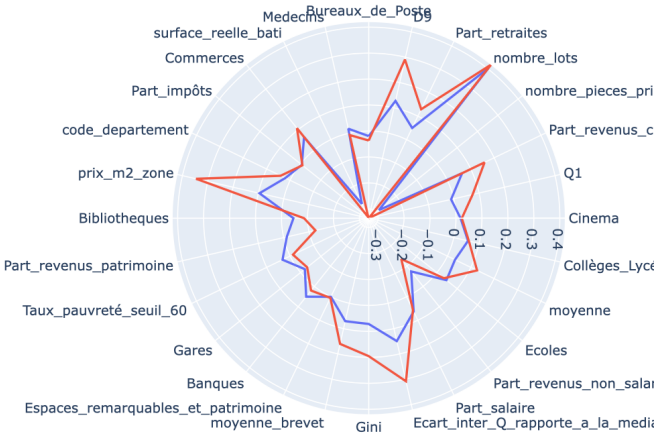
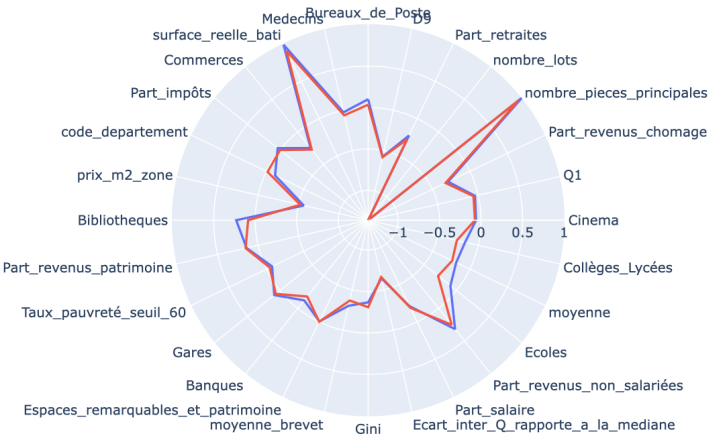



Figure 16: Histogram of errors made by the Nice-Apartment model

The models generally have error histograms similar to this one: a large majority of errors are less than 50%, then the number of errors decreases rapidly but we continue to make bad predictions, sometimes up to 150%. An interesting question would be to ask if the properties that lead to these high errors (>100%) have significantly different characteristics from other properties.




(a) Spider chart displaying the average characteristics of two groups: individuals with low errors in blue (113316 individuals), individuals with high errors in red (290253 individuals) **for apartments**

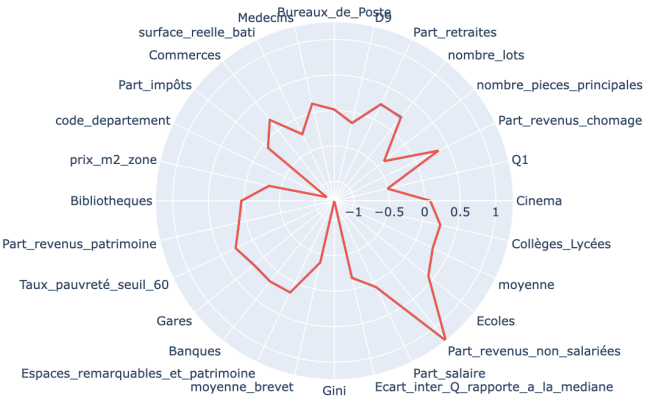



(b) Spider chart displaying the average characteristics of two groups: individuals with low errors in blue (88290 individuals), individuals with high errors in red (33171 individuals) **for houses**

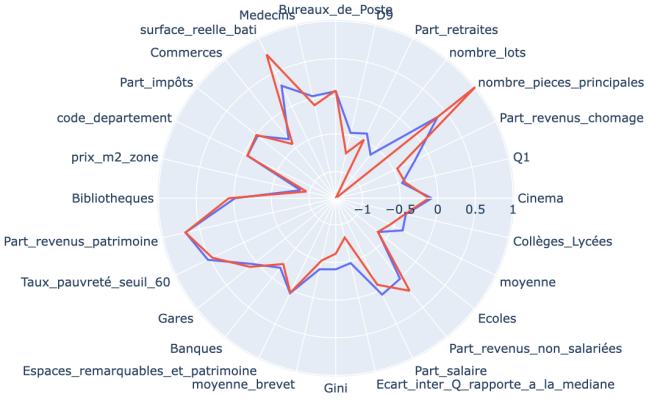
We notice that for apartments, the properties with high errors more often have a high price per square meter of the neighborhood. This may be related to the fact that our model overfits the price of the nearest neighbors, and if this price is high but our property is in a worse condition, it

can potentially cost much less, an effect not captured by our model which then overestimates the price, leading to larger errors.

On the other hand, for houses, the profiles of the two groups are very similar: this shows that there is no fundamental difference between the two groups. This seems to suggest that our models suffer from common variable biases and that we do not have enough indicators to effectively differentiate properties with different prices per square meter.




(a) Spider chart displaying the average characteristics of two groups: individuals with low errors in blue (22076 individuals), individuals with high errors in red (11239 individuals) **in Nice**




(b) Spider chart displaying the average characteristics of two groups: individuals with low errors in blue (13722 individuals), individuals with high errors in red (4662 individuals) **in Montpellier**

We also observe this phenomenon for cities, where we can see that the profiles between the two groups are generally very similar. Despite a few cities where differences are observed, most curves are very close, which once again suggests that the errors are the result of unexplained factors rather than directly related to a modeling or variable error in our model.There are still some cities where significant differences seem to emerge, such as Montpellier, where certain variables such as the actual surface area of the building, the number of main rooms or the number of lots seem to have an effect on the model's strong errors. We could then try to display these variables by IRIS to try to understand if it is a geographical specificity, an error in the processing of variables, an inconsistency in the base dataset...However, in most cities, such as Nice where we still make a number of large errors, there is no significant trend that seems to emerge between the groups, showing that our variables are still incomplete in capturing the full reality of the real estate markets in each metropolis.

Now that we have a better understanding of the model's errors, an important question remains: will our models be able to perform well when we don't provide them with a discount rate? In other words, will our models continue to be as accurate when deployed in the real world?

## 5.3 Robustness of models over time

As previously explained in the constraints of this project, temporality plays an important role. Meilleur Taux wishes to obtain a model that is as robust as possible to time. We therefore decide to analyze the performance of our model over time. To do this, we compare the predictions of the models with these same predictions updated. Indeed, the current models predict the price per $m^2$ of a property in the second quarter of 2021. So we update these prices to their real quarter of sale. To do this, we use the technique explained at the beginning of this report.

The results obtained for each metropolis are presented in the table below:

| **Apartment** | | |
|---|---|---|
| **Metropole** | **diff MeAPE** | **diff MAPE** |
| Paris | 0,04% | -0,04% |
| Nice | 0,22% | 0,80% |
| Marseille | 0,33% | 0,86% |
| Rennes | -2,41% | -1,40% |
| Toulouse | 0,16% | 0,99% |
| Bordeaux | 1,26% | 1,97% |
| Montpellier | 0,07% | 0,21% |
| Nantes | -1,16% | -0,30% |
| Lille | 0,07% | 0,01% |
| Lyon | 0,05% | 0,02% |
| **Mean** | **-0,13%** | **0,31%** |

| **House** | | |
|---|---|---|
| **Metropole** | **diff MeAPE** | **diff MAPE** |
| Paris | -0,24% | 0,02% |
| Nice | -0,30% | 0,68% |
| Marseille | -0,09% | 1,16% |
| Rennes | -1,48% | -0,70% |
| Toulouse | 0,73% | 1,04% |
| Bordeaux | -0,05% | 1,13% |
| Montpellier | -1,08% | -0,27% |
| Nantes | -1,11% | -0,08% |
| Lille | -0,55% | 0,37% |
| Lyon | -0,92% | -0,45% |
| **Mean** | **-0,51%** | **0,29%** |

We notice that for all the metropolises and all the properties (apartments and houses) discounting does not significantly improve the results. The differences between the results are minimal. Indeed, on average, for apartments, updating reduces the median percentage error by only 0.13% and by 0.51% in the case of houses. On the contrary, the discounting increases the median of the percentage of error by 0.3%. We conclude that the model seems to be robust to the near temporality. Indeed, the tested data ran for one year after the update date of the models train set (Q2-2021). Thus, the requirement of robustness of the predictions to the temporality required by Meilleur Taux is correctly satisfied.

# Conclusion

## A look back to our objectives

Our work provides a ready to use tool to predict real estate price in the most important french metropolises. We have payed attention to the main challenges we had identified. The temporal component has been handled with price actualisation. We have tackled data leakage challenge with rigorous train test split and preprocessing taking into account the temporal dimension of train and test data. Real estate being all about location we have splited our work on the metropolises. This part could be improved by adding some variables specific to the metropolises. This geographical split implies that we have inequal results through metropolises but this also allows us to take the most of some information in metropolises. We therefore have some very good results for some of them. Future improvements have a solid basis for development.

## Limits and potential improvements

We have identified few limits and possible improvement to our project.

- We could have benefited from some aditional information about the properties. This would include for example : number of bedrooms, number of bathrooms, information about energy consumption of the property (energy class A-G), state of the property or date since last work. Other variables could be more specialy linked to each metropolis. For example the sea view for Nice. This would potentially help reduce overfitting on *price_ zone*.
- We have used random gridsearch so we don't test for all hyperparameters. Maybe the best ones have not been tested.
- We could have refined our work on the *price_ zone* for example by adding some informations about the distance.
- The price actualisation indices is based on an quite blurred methodology. Moreover, only four metropolis have their own indices. For all the others, the indice is less precise.

## Project utilization

The project was transformed into a package comprising various modules, namely pre-processing, machine learning, and EDA. These modules are invoked from a main function and generate outputs such as data, plots, metrics, and models, which are stored in a local folder. You can find the project on **GitHub** at the following link.For instructions on how to use the project, please refer to the README.md file. Additionally, the project also offers extensive documentation for all modules, the link to which is provided in the README.

# Bibliography


1. Fagereng, Blomhoff Holm, Torstensen, *"Housing Wealth in Norway 1993-2015"* - 2020.

2. Joshua H. Gallin, Raven Molloy, Eric Nielsen, Paul Smith, andKamila Sommer, *"Measuring Aggregate Housing Wealth: New Insights from anAutomated Valuation Model"* - 2018.

3. Céline Grislain-Letrémy, Arthur Katossky, *"he impact of hazardous industrial facilities on housing prices: Acomparison of parametric and semiparametric hedonic price models"* - 2014.

4. Katharina Knoll, Moritz Schularick, Thomas Steger, *"No Price Like Home: Global House Prices, 1870–2012"* - 2017.

5. Sendhil Mullainathan, Jann Spiess, *"Machine Learning: An Applied Econometric Approach"* - 2017.

6. Mathilde Poulhes, *"From Latin Quarter to Montmartre Investigating Parisian Real-Estate Prices"* - 2017.

7. Quang Truong, Minh Nguyen, Hy Dang, Bo Mei, *"Housing Price Prediction via Improved Machine Learning Techniques"* - 2020.

8. Arnaud Hureaux- Projet : Estimateur de prix d'un bien immobilier basé sur du Machine Learning : `https://hureauxarnaud.medium.com/projet-estimateur-de-prix-dun-bien-immobilier-base-sur-du-machine-learning-ae578fdacaca/`.

9. Nada Belaidi- XGBoost : Tout savoir sur le Boosting : `https://blent.ai/xgboost-tout-comprendre/`.

10. Vianney Perchet, *"Theoretical fundations of Machine Learning"* (Cours de deuxième année de l'ENSAE).

# Appendix

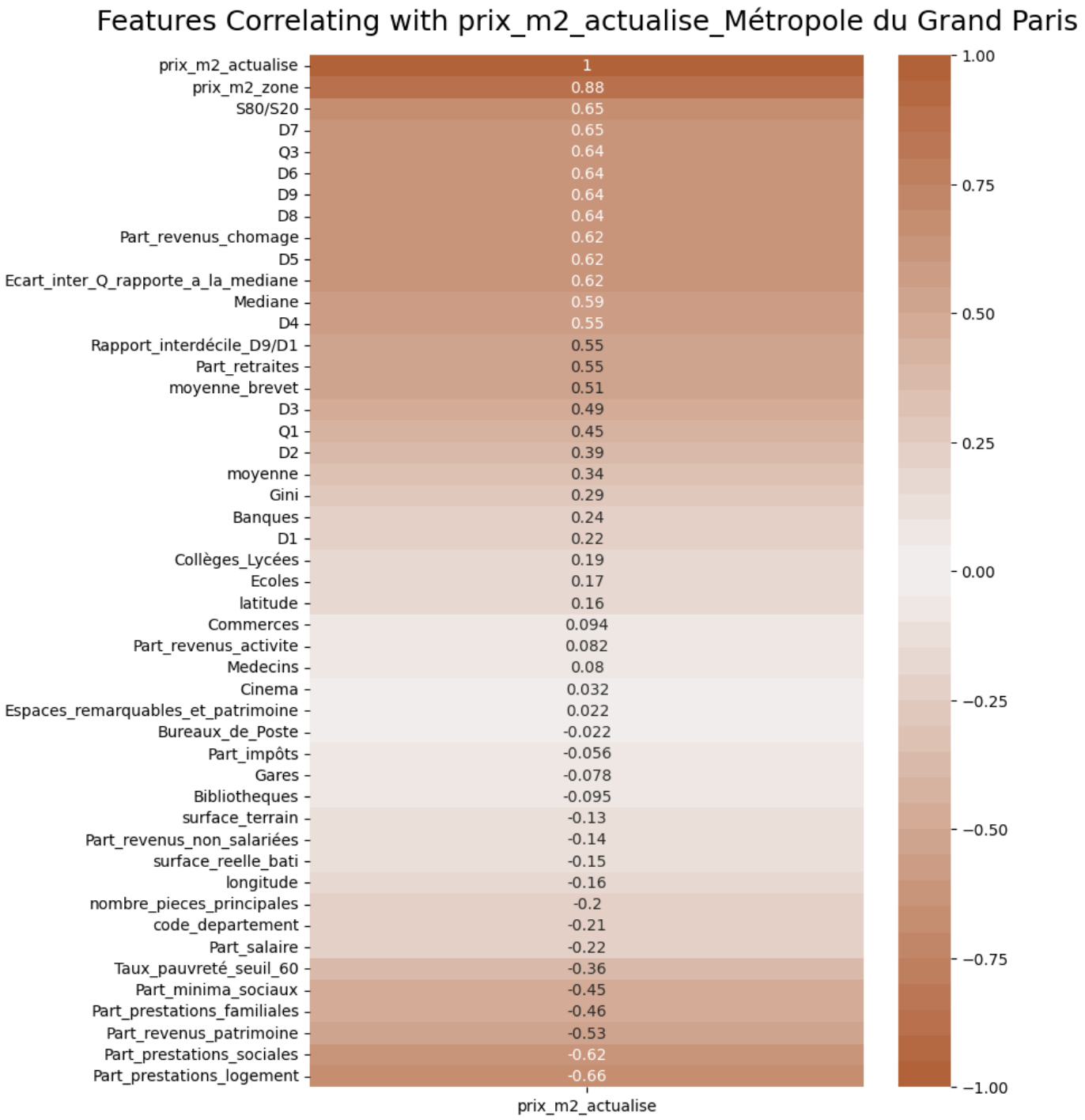



Figure 19: Feature correlation with prix_$m^2$_actualise Grand Paris Métropole

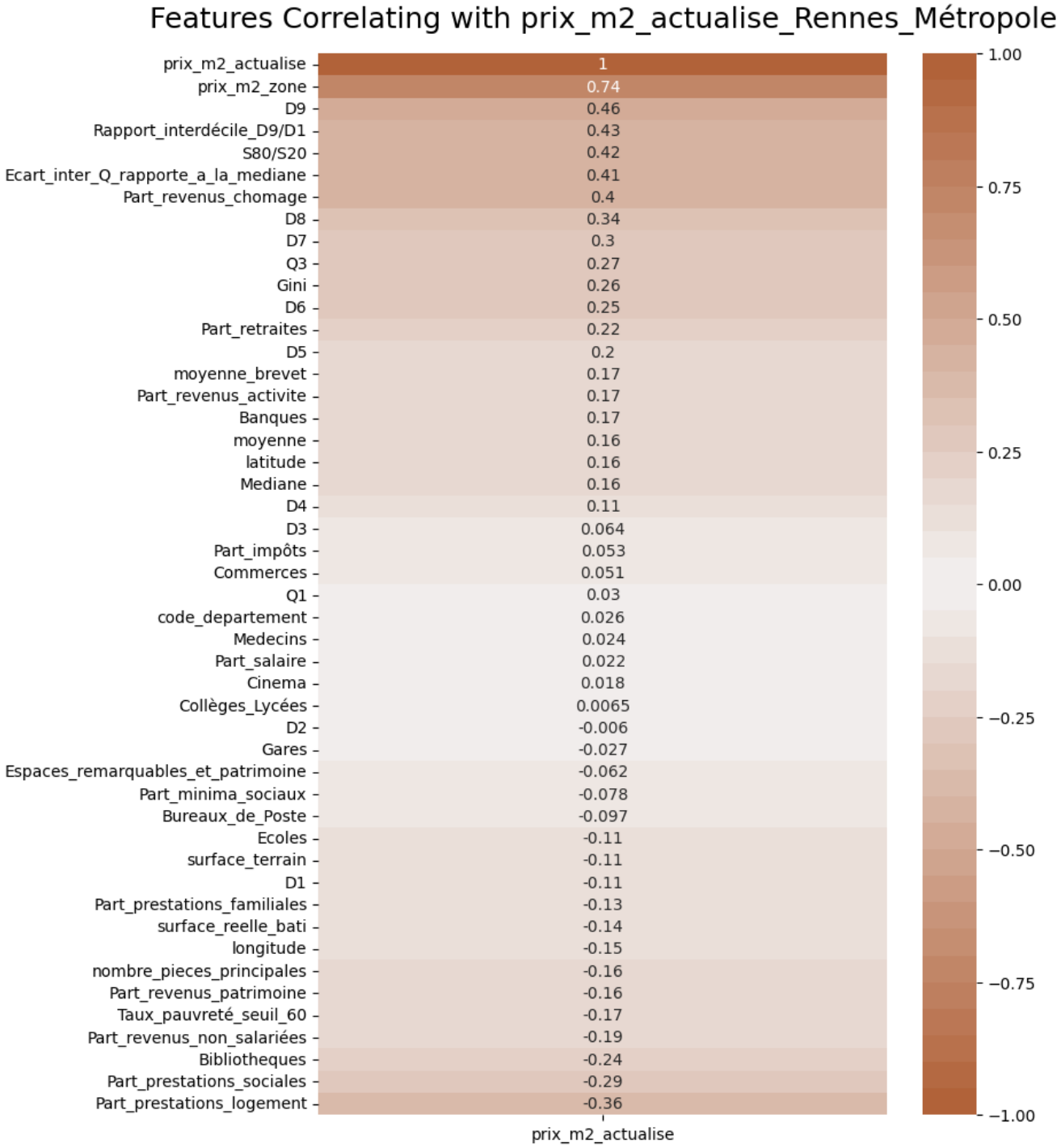



Figure 20: Feature correlation with prix_$m^2$_actualise Rennes Métropole

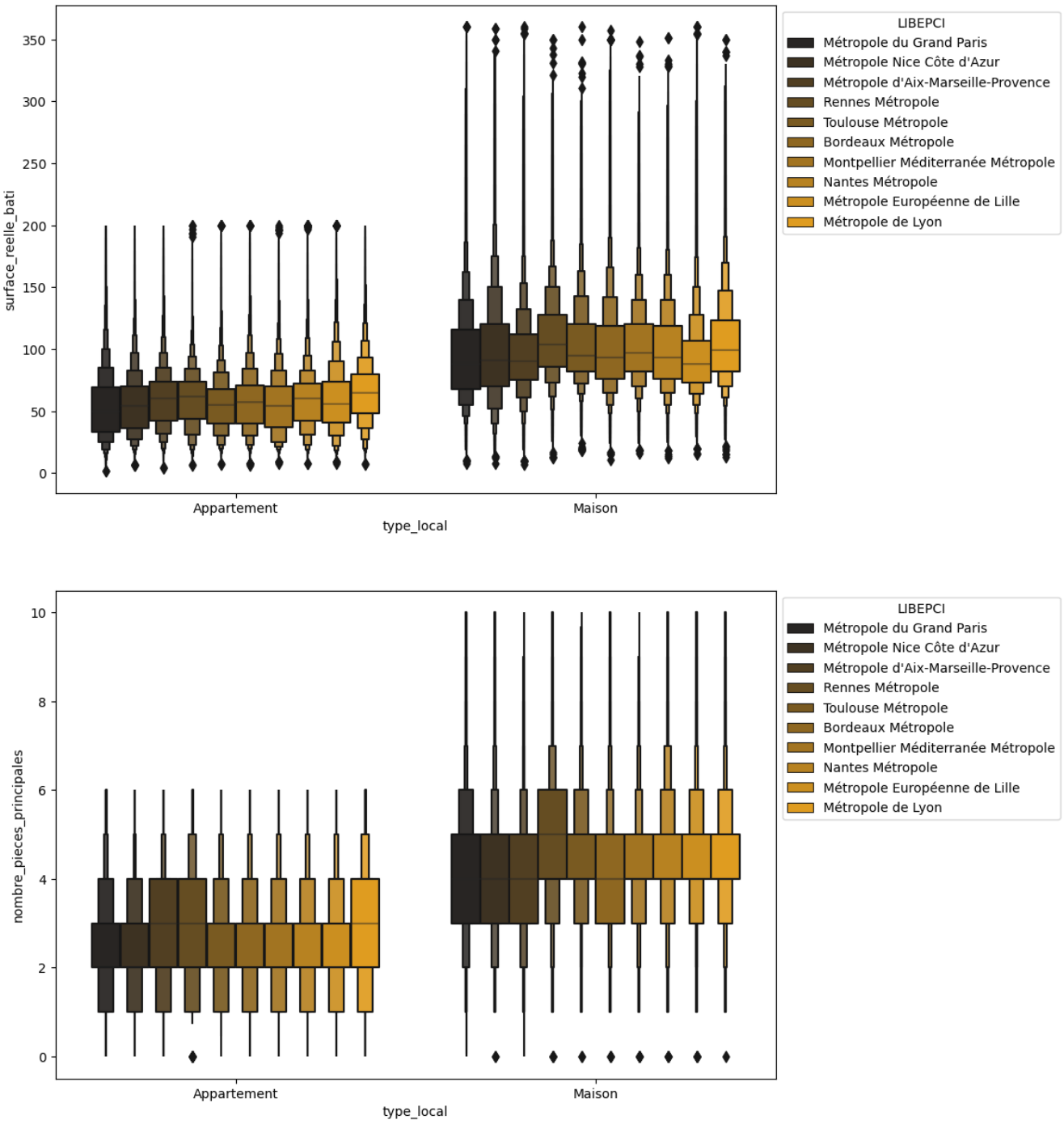



Figure 21: Distribution of surface and number of rooms by type of properties, by metropole

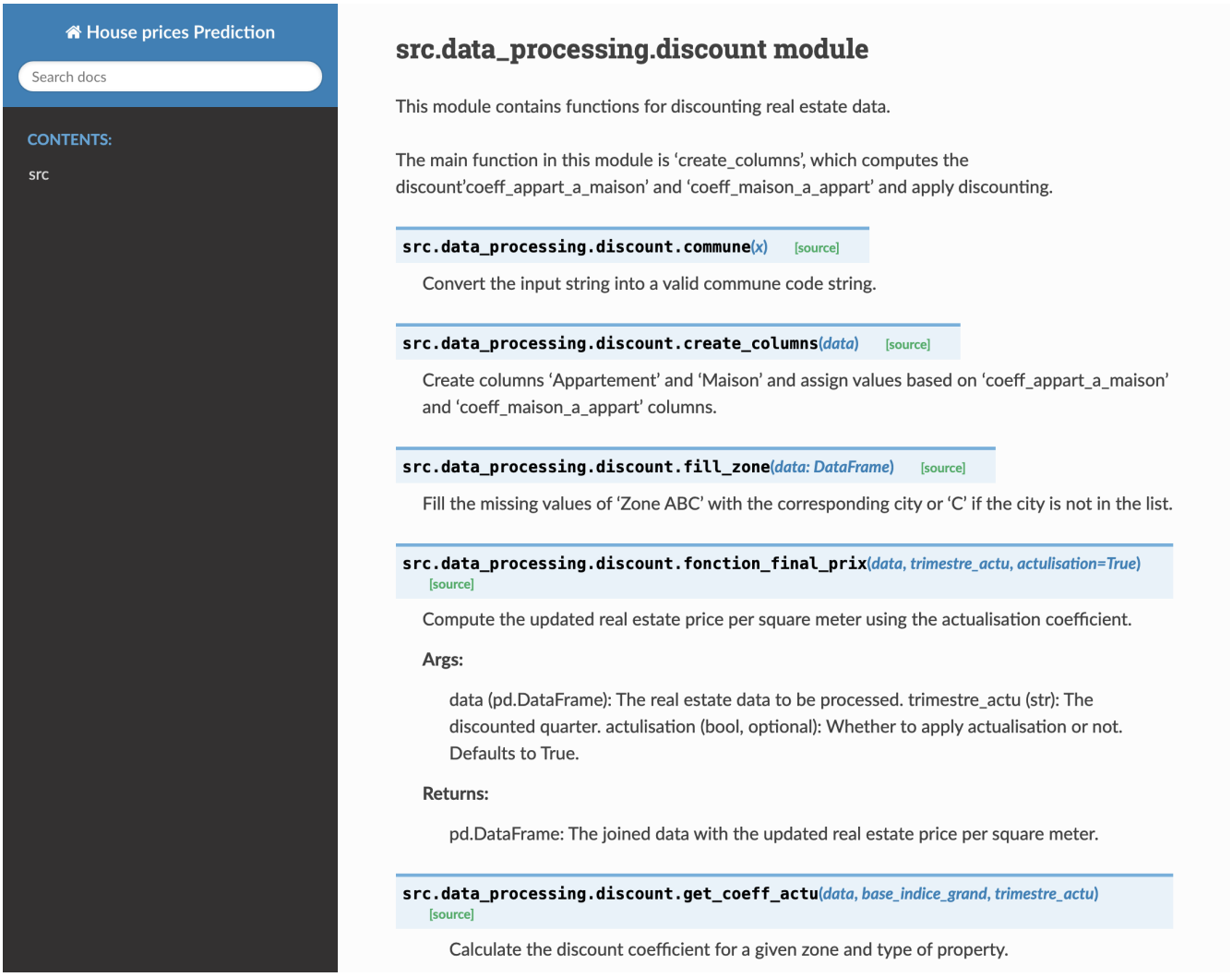



Figure 22: Documentation page of 'Discount' module

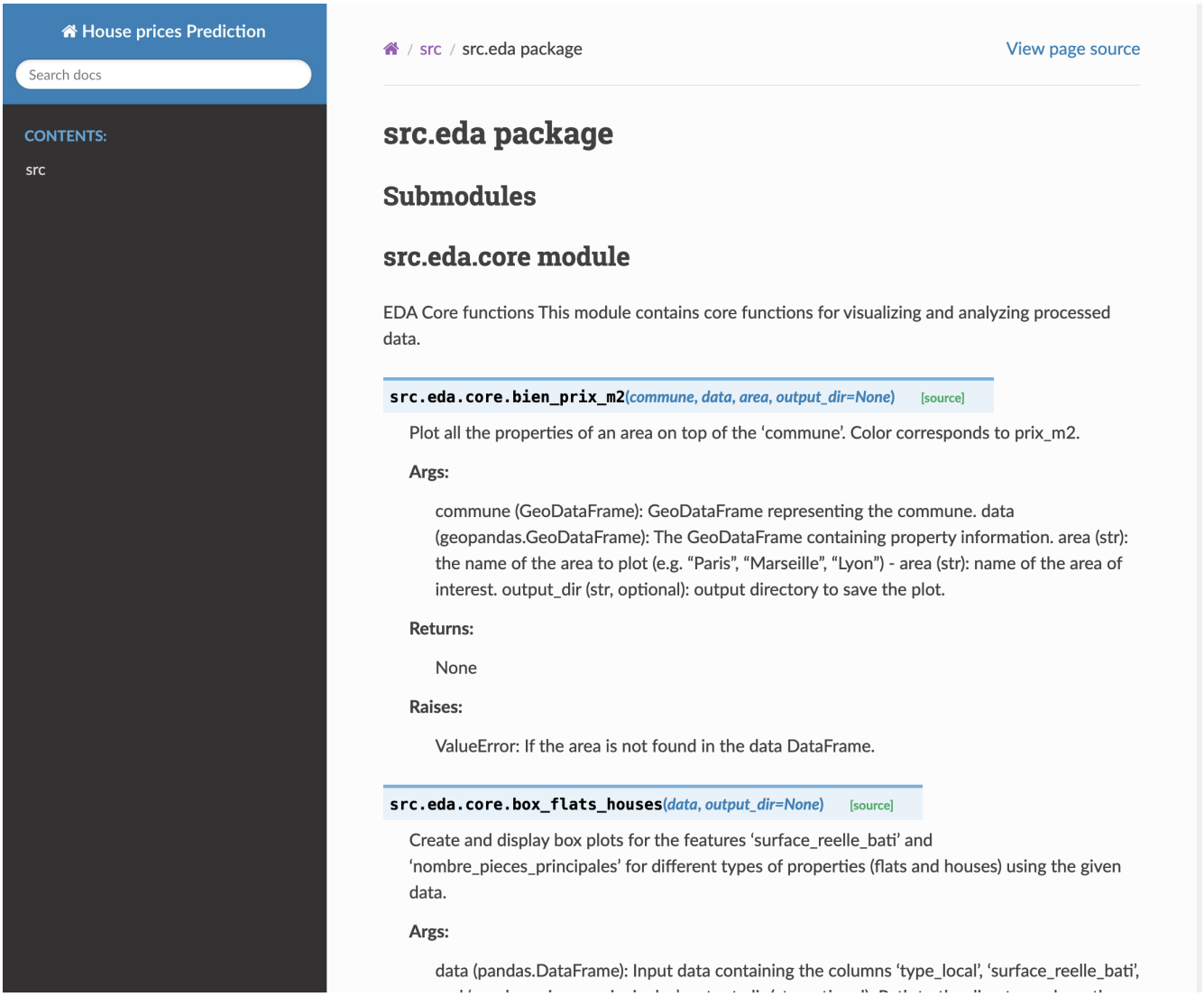



Figure 23: Documentation page of 'Core' module