Université de Lille , Sciences Humaines et Sociales
UFR Mathématiques, Informatique, Management, Économie

Algorithmes fondamentaux de la fouille de données

# Le prix Zillow : Prévision de la valeur de bien immobilier de Zillow (Zestimate)


**Préparé par :**

Ben Yassine Mohamed

Ellyes Khalfaoui

Imen Rezoug

# Table des matières

# Première partie

# Introduction

Une maison est souvent l'achat le plus important et le plus coûteux qu'une personne fait au cours de sa vie. Il est extrêmement important de s'assurer que les propriétaires de maison ont un moyen fiable de surveiller cet actif.
Le Prix Zillow, un concours doté d'un grand prix d'un million de dollars, met au défi la communauté scientifique des données de contribuer à pousser encore plus loin la précision de la Zestimate. Les algorithmes gagnants auront un impact sur la valeur de 110 millions de maisons aux États-Unis.

Le Zestimate a été créé pour donner aux consommateurs le plus d'information possible sur les maisons et le marché de l'habitation. C'est la première fois que les consommateurs ont accès gratuitement à ce type d'information sur la valeur des maisons.

Les " Zestimate " sont des valeurs estimées des maisons basées sur 7,5 millions de modèles statistiques et d'apprentissage automatique qui analysent des centaines de points de données sur chaque propriété. Et, en améliorant continuellement la marge d'erreur médiane (de 14 % à l'origine à 5 % aujourd'hui), Zillow s'est depuis établi comme l'une des places de marché les plus importantes et les plus fiables pour les informations immobilières aux États-Unis et un exemple de premier plan de l'apprentissage automatique percutant.

Le but de ce modèle est d'améliorer l'erreur résiduelle de Zestimate. Plus précisément, nous essayons de minimiser l'erreur absolue moyenne entre l'erreur logarithmique prédite et l'erreur logarithmique réelle. Cette information est enregistrée dans les données de formation des transactions.

$$logerror = log(Zestimate) - log(SalePrice)$$

# Deuxième partie

# Exploration des Données

## 0.1 Description des données et du problème :

### 0.1.1 Les datasets :

Le dataset d'entrainement contient les transaction avant le 15 octobre 2016 et des transaction après le dataset de test contient les transactions entre le 15 et 31 décembre,le reste du dataset de test est utilisé pour faire le classement des participants et contient toutes les propriétés entre le 15 octobre et le 15 décembre 2017.

### 0.1.2 Les fichiers :

**properties_2016.csv** : Les propriétés avec leurs caractéristiques pour l'année 2016, avec des propriété de l'année 2017 avec un manque de valeurs dans les caractéristiques.

**properties_2017.csv** : Les propriétés avec leurs caractéristiques en 2017.

**train_2016.csv** : l'ensemble d'entrainnement avec les transactions entre le 1er janvier 2016 et le 31 décembre 2016.

**train_2017.csv** : l'ensemble d'entrainnement avec les transactions entre le 1er janvier 2017 et le 15 septembre 2017.

**sample_submission.csv** : un échantillion example de la solution à soumettre zillow_data_dictionary.

## 0.2 Étude exploratoire

Dans la partie pré-traitement, on a commencé d'abord par regarder notre jeu de données, voir quel type de problème pourrait être lié par rapport à ces données, comprendre ce que représente chaque caractéristique (attributs), d'abord par un aperçu des premières lignes du dataset, puis les statistiques descriptives (mean, std..) pour chaque attribut, nous avons ensuite regardé le domaine des instances, puis pour des raisons de lisibilité, nous avons recodé les noms des attributs, après avoir lu le dictionnaire joint au dataset, nous avons ensuite regardé les valeurs nulles (NaN) et leurs proportions dans le dataset, nous avons pris la décision d'exclure certains attributs du traitement si le taux est proche de 100 % de valeurs nulles, nous avons remplacé les valeurs manquantes des autres attributs par la valeur la plus représentée (**mod**), on a vu d'autres méthodes pour le traitement des valeurs manquantes, dans un travail d'une autre personne, ils ont utilisé K-nn pour trouver

les valeurs manquantes...
Nous avons ensuite combiné deux fichiers pour avoir notre dataset d'entrainement (ajoute du fichier de log-error correspondant à l'ID de la propriété), nous avons ensuite fait des visualisations afin de voir la pertinence des attributs, et notamment en associant le log-error qui représente la donnée cible du problème avec d'autres attributs, comme par exemple l'évolution du log-error (valeur absolue) au fil des mois pour l'année 2016.

Nous avons ensuite visualisé les corrélations entre les attributs (**Corrélation de Pearson**) afin de déterminer les attributs pertinents. Nous avons également transformé les valeurs ordinales en valeurs quantitatives pour certains attributs avant de réaliser une normalisation des données (afin que les résultats de modèles statistiques ne soient pas biaisés par la différence d'échelle des valeurs des différents attributs).

## 0.3 A.C.P :

Avant d'analyser les résultats proprement dits d'une A.C.P., il est bon d'en regarder les **résultats préliminaires**. Tout d'abord, pour chaque variable considérée, son minimum, son maximum, sa moyenne et son écart-type. Cela permet d'avoir une première connaissance des données étudier et le car échéant, de décider si L'A.C.P. doit être réduite ou non.
Il est également intéressant d'étudier la **matrice de corrélation** entre variables initiales, dans la mesure ou elle permet d'avoir une première idée de la structure de corrélation entre ces variables.

## 0.4 Statistiques descriptives

| | aircon | architectural_style | area_basement | num_bathroom | \ |
|---|---|---|---|---|---|
| count | 90275.000000 | 90275.000000 | 90275.000000 | 90275.000000 | |
| mean | 1.260271 | 7.000665 | 1527.612074 | 2.279474 | |
| std | 1.721860 | 0.146291 | 20.119940 | 1.004271 | |
| min | 1.000000 | 2.000000 | 100.000000 | 0.000000 | |
| 25% | 1.000000 | 7.000000 | 1528.000000 | 2.000000 | |
| 50% | 1.000000 | 7.000000 | 1528.000000 | 2.000000 | |
| 75% | 1.000000 | 7.000000 | 1528.000000 | 3.000000 | |
| max | 13.000000 | 21.000000 | 1555.000000 | 20.000000 | |

| | num_bedroom | framing | quality | num_bathroom_calc | deck | \ |
|---|---|---|---|---|---|---|
| count | 90275.000000 | 90275.0 | 90275.000000 | 90275.000000 | 90275.0 | |
| mean | 3.031869 | 4.0 | 6.088408 | 2.305168 | 66.0 | |
| std | 1.156436 | 0.0 | 1.664972 | 0.970398 | 0.0 | |
| min | 0.000000 | 4.0 | 1.000000 | 1.000000 | 66.0 | |
| 25% | 2.000000 | 4.0 | 4.000000 | 2.000000 | 66.0 | |
| 50% | 3.000000 | 4.0 | 7.000000 | 2.000000 | 66.0 | |
| 75% | 4.000000 | 4.0 | 7.000000 | 3.000000 | 66.0 | |
| max | 16.000000 | 4.0 | 12.000000 | 20.000000 | 66.0 | |

| | tax_building | tax_total | tax_year | tax_land | tax_property | \ |
|---|---|---|---|---|---|---|
| count | 9.027500e+04 | 9.027500e+04 | 90275.0 | 9.027500e+04 | 90275.000000 | |
| mean | 1.797563e+05 | 4.576714e+05 | 2015.0 | 2.783325e+05 | 5983.680847 | |
| std | 2.087537e+05 | 5.548814e+05 | 0.0 | 4.004942e+05 | 6838.745460 | |
| min | 1.000000e+02 | 2.200000e+01 | 2015.0 | 2.200000e+01 | 49.080000 | |
| 25% | 8.149000e+04 | 1.990235e+05 | 2015.0 | 8.222750e+04 | 2872.470000 | |
| 50% | 1.315070e+05 | 3.428720e+05 | 2015.0 | 1.929600e+05 | 4542.440000 | |
| 75% | 2.100425e+05 | 5.405890e+05 | 2015.0 | 3.454150e+05 | 6900.600000 | |
| max | 9.948100e+06 | 2.775000e+07 | 2015.0 | 2.450000e+07 | 321936.090000 | |

| | tax_delinquency_year | censustractandblock | logerror |
|---|---|---|---|
| count | 90275.000000 | 9.027500e+04 | 90275.000000 |
| mean | 13.988203 | 6.049076e+13 | 0.068447 |
| std | 0.390536 | 2.041793e+11 | 0.146262 |
| min | 6.000000 | 6.037101e+13 | 0.000000 |
| 25% | 14.000000 | 6.037400e+13 | 0.013900 |
| 50% | 14.000000 | 6.037620e+13 | 0.032500 |
| 75% | 14.000000 | 6.059042e+13 | 0.069400 |
| max | 99.000000 | 6.111009e+13 | 4.737000 |

[8 rows x 53 columns]

## 0.5 Matrice de corrélation

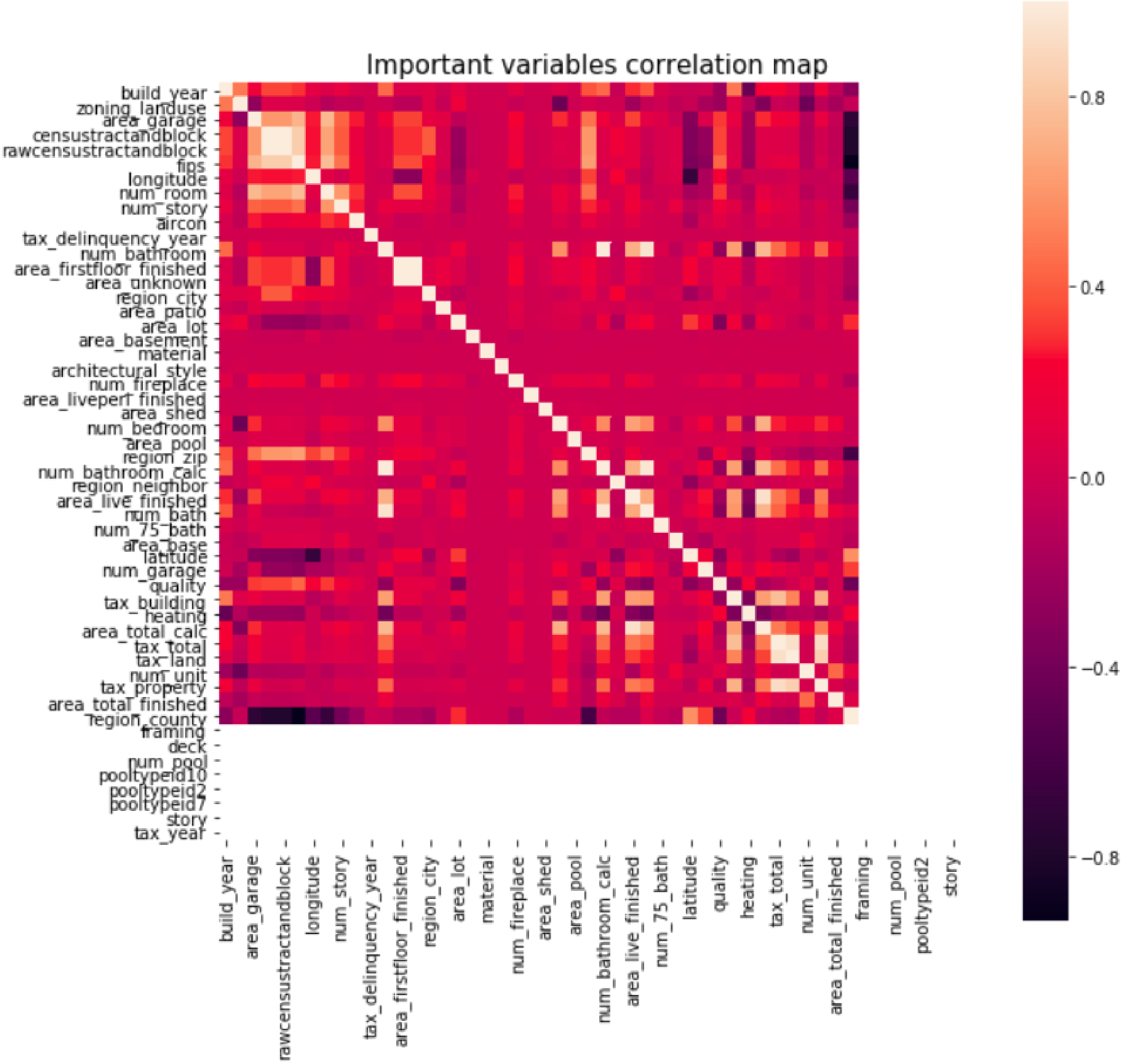



**Figure : Matrice de corrélation de l'ensemble des variables**

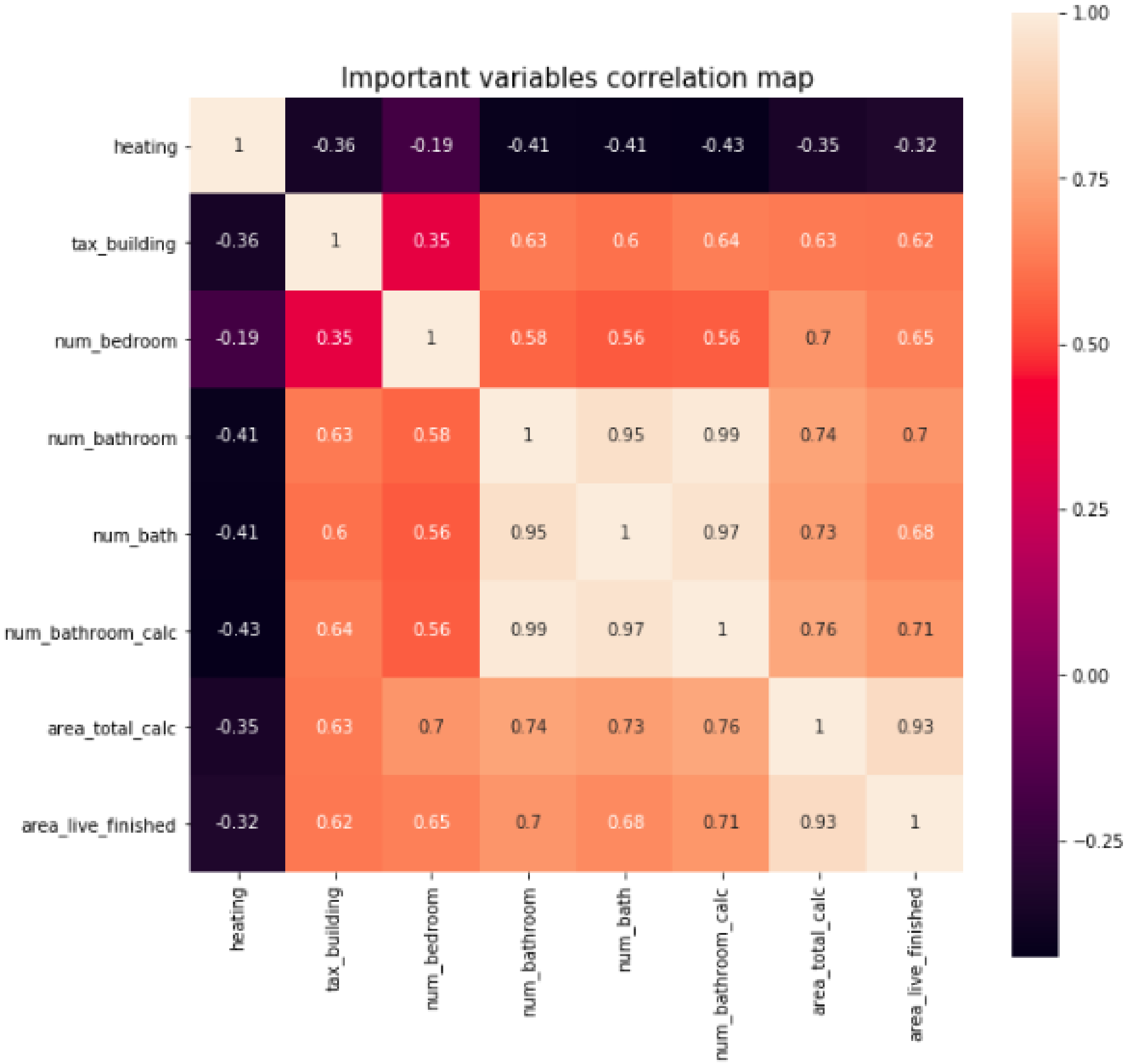



**Figure : Matrice de corrélation des variables pertinantes**

C'est la matrice de variance covariance des variables réduites. Elle possède p valeurs propres. Le coefficient de corrélation nous donne deux informations que l'on doit interpréter :

**-Le sens de relation entre les variables :** si le coefficient est négatif, plus la valeur de la première variable est élevé, plus la valeur de la deuxième diminue.

**- La force de la relation :** En examinant la valeur de chaque coefficient, nous pouvons dire que l'effet de la relation entre deux variables est de grande taille et que l'association est très forte, ou bien le contraire.
On remarque que le coefficient de corrélation entre la variable "**num_bathroom_calc**
" et la variable "**num**$_{b}atroom" estpositive, donclesdeuxvariablessontpositivementcorrles,$

# Troisième partie

# Applications des Méthodes

# Chapitre 1

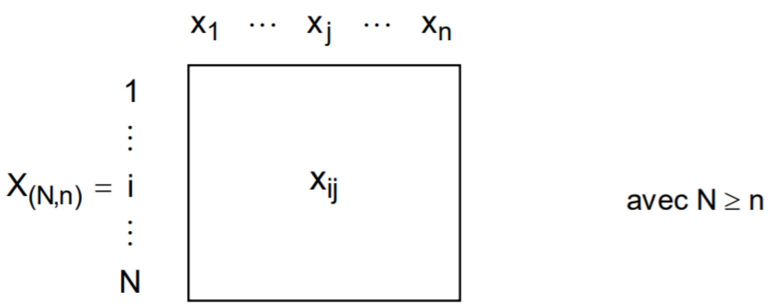
# Analyse en compostantes principales

L'objectif de l'Analyse en Composantes Principales (ACP) est de revenir à un espace de dimension réduite (par exemple 2) en déformant le moins possible la réalité.

Il s'agit donc d'obtenir le résumé le plus pertinent possible des données initiales. C'est la matrice des variances-covariances (ou celle des corrélations) qui va permettre de réaliser ce résumé pertinent, parce qu'on analyse essentiellement la dispersion des données considérées.
De cette matrice, on va extraire, par un procédé mathématique adéquat, les facteurs que l'on recherche, en petit nombre.

## 1.1 Formalisation mathématique de l'ACP :

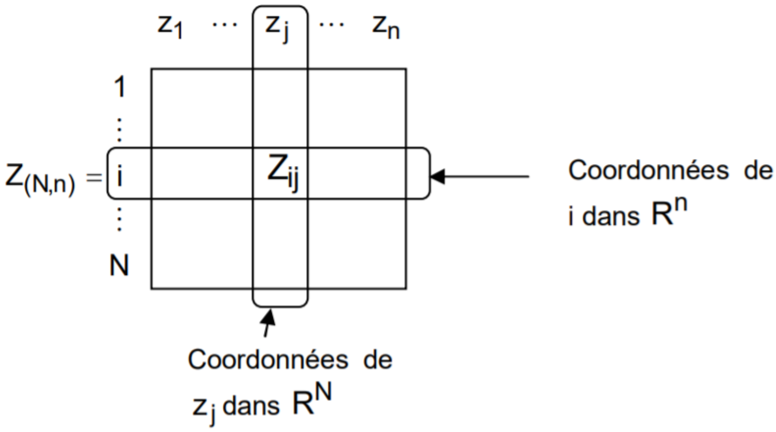
Le tableau de départ qui sera soumis à une ACP se présente de la façon suivante :




L'idée de l'ACP est de déterminer un nouveau repère de $R^p$ associé de manière naturelle à la structure du nuage considéré, de façon à pouvoir l'y examiner plus commodément. L'ACP vise donc à projeter dans $R^n$ orthogonalement les N points individus sur n nouveaux axes appelés **axes principaux**, sachant que l'origine de ces nouveaux axes reste identique à celui de l'espace de départ. **Centrer et réduire** Mathématiquement cela revient à soustraire la moyenne pour chaque variable et diviser par son écart type et on obitient la nouvelle variable Z :

$$Z = \frac{X - \mu}{\sigma}$$

Pour s'affranchir des effets d'échelle dus à l'hétérogénéité éventuelle des variables, ces dernières sont en général normalisées, c'est à dire que chaque colonne est divisée par son écart-type, toutes sont dès lors exprimées dans la même échelle standard. D'autre part, l'origine est placée au centre de gravité du nuage. C'est le nuage ainsi est en fait considéré. On obtient alors La Matrice $Z$ des variables centrées réduites qui s'écrit sous la forme suivante :




L'information contenue dans cette matrice est donnée par le nuage de points des individus dans l'espace $R^N$.
Plaçons nous dans l'espace $R^n$ des variables qui contient le nuage des $N$ points individus. Le système des n axes est orthonormée ou encore la base de ce système est orthonormée. L'information contenue dans ces espaces est illisible du fait du nombre d'axes. Après avoir transformé la matrice initial à une matrice centrée réduite, on se place dans l'espace $R^n$ avec un système orthonormé. On analyse le nuage des $N$ points individus. On calcule la matrice de corrélation entre les variables, c'est une matrice carrée symétrique.

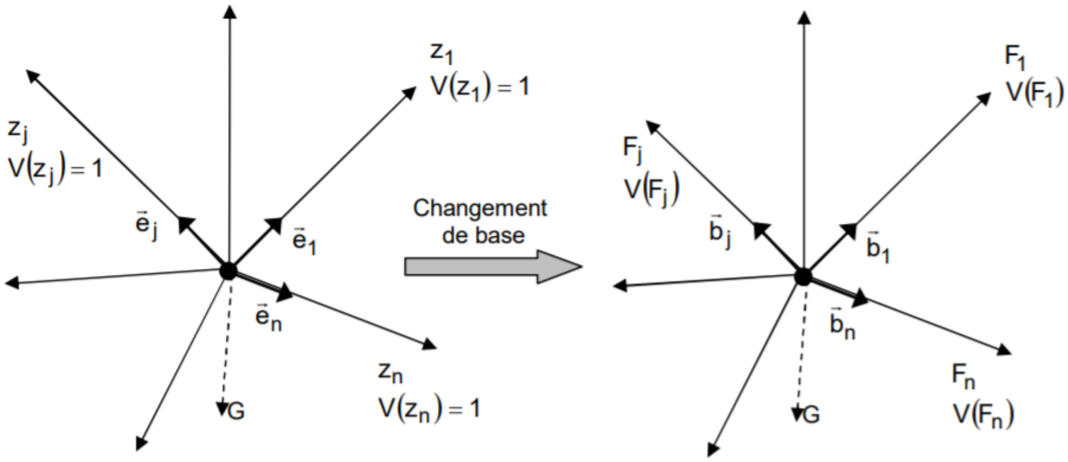
On diagonalise **R**, c'est à dire qu'on calcule les **n** valeurs propres de cette matrice, on divise chacune des valeurs par **n**, ce qui donne le pourcentage de variance expliquée par une composante principale. puis on calcule les vecteurs propres appellées (composante principale) On cherche des **combinaisons linéaires** des variables initiales, appelées **facteurs**, ou encore **composantes principales**, s' écrivant sous la forme suivante :

$$C^1 = a_1^1 X^1 + a_1^2 X^2 + ... + a_1^p X^p$$
$$C^2 = a_2^1 X^1 + a_2^2 X^2 + ... + a_2^p X^p$$

telles que : $C^1$ doit contenir un maximum d'information, c'est à dire dispenser le plus possible les individus. L'idée est la suivante :
Si on dispose d'un nuage de points dans le plan (autrement dit, en dimension $p = 2$) et qu'on souhaite le projeter sur une droite (donc eb dimension $q = 1$), la droite la plus "fidèle" à la configuration initiale est celle qui rend maximum la dispersion

la variance du nuage après sa projection. Le critère choisi est, de façon naturelle, var($C^1$) maximum. Il s'agit des outils de l'algèbre linéaire, essentiellement les notions de **vecteurs propres** et de **valeurs propres.** Notons **S** la matrice $pxp$ des variances-covariances des variables $X^j$ et **R** la matrice $pxp$ de leurs corrélations linéaires. Dans une A.C.P. seulement centée, $C^1$ est le vecteur propre normé de **S** associé à la plus grande valeur propre ( $\mathbf{S}C^1 = \lambda\ C^1$ et $\|C^1\| = 1$ ), $C^2$ est le vecteur propre normé de **S** associé à la seconde plus grande valeur propre, et ainsi de suite. De plus, les différents vecteurs $C^k$ sont orthogonaux (à la non corrélation des variables centrées correspond l'orthogonalité des vecteurs qui les représentent). Dans une A.C.P. réduite, les $C^k$ sont les vecteurs propres orthonormés de la matrice **R**.

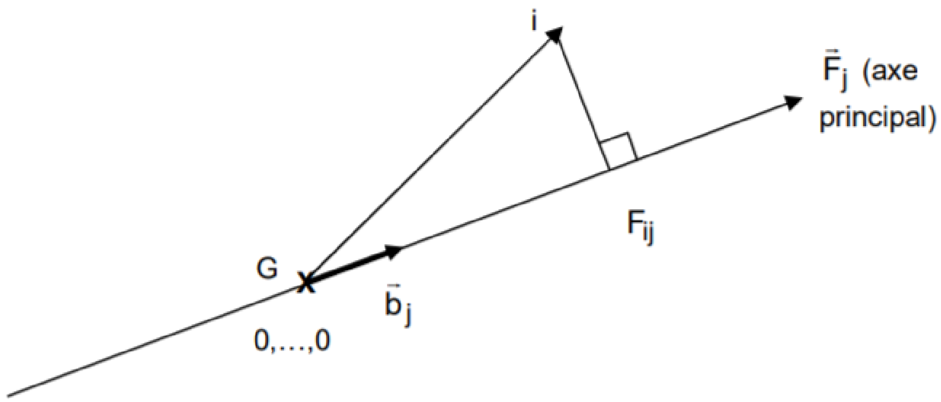
Le principe de l'ACP consiste donc à effectuer dans $R^n$ et dans $R^N$ un changement de base de telle sorte (lorsque cela est possible) que les variances des projections orthogonales (les coordonnées) sur les nouveaux axes (appelés axes principaux) rassemblent une part significative de la variance totale à partir des deux ou trois premiers axes. On peut schématiser ce principe de la façon suivante dans $R^n$ .




La variance des coordonnées des N points individus sur $z_j$ (quelquesoit j), explique $\frac{1}{n}$ * 100 de la variance totale.

Système orthonormé d'axes principaux $\vec{F}_j$ (quelquesoit j), de même origine, de base orthonormée : $(\vec{b}_1,\ldots\vec{b}_n)$ et tel que la variance des coordonnées de N points individus sur $\vec{F}_1,\vec{F}_2$ et au maximum $\vec{F}_3$ représente, par exemple, 70% à 80% de la variance totale n.

Rappelons que si on connaît les coordonnées d'un vecteur quelconque $\vec{b_j}$ dans la base $R^n$ de départ, la projection orthogonale $F_{ij}$ (la coordonnée) d'un point i du nuage des $N$ points est donnée par le produit scalaire du vecteur $\vec{b_j}$ par le vecteur $\vec{G}\,\vec{i}$ ou $G$ est l'origine des axes ($G$ est le centre de gravié du nuage des $N$ points) :




Après avoir transformé la matrice initial en une matrice centrée réduite, on se place dans l'espace $R$.

## 1.2 Application de l'ACP sur le jeu de données en Python :

**Normalisation des données** La première étape pour appliquer l'analyse en composante principale est de normaliser les variables. Pour faire cela sur python il existe la librairie suivante :

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data =scaler.fit_transform(X)
```

```
             aircon  architectural_style  area_basement  num_bathroom  \
count  9.027500e+04         9.027500e+04   9.027500e+04  9.027500e+04
mean  -4.565104e-17         5.284182e-16   5.603021e-15  1.760713e-16
std    1.000006e+00         1.000006e+00   1.000006e+00  1.000006e+00
min   -1.511579e-01        -3.418318e+01  -7.095548e+01 -2.269792e+00
25%   -1.511579e-01        -4.543269e-03   1.928077e-02 -2.782868e-01
50%   -1.511579e-01        -4.543269e-03   1.928077e-02 -2.782868e-01
75%   -1.511579e-01        -4.543269e-03   1.928077e-02  7.174659e-01
max    6.818087e+00         9.569563e+01   1.361240e+00  1.764526e+01

        num_bedroom  framing       quality  num_bathroom_calc     deck  \
count  9.027500e+04  90275.0  9.027500e+04       9.027500e+04  90275.0
mean   5.399416e-17      0.0  2.110180e-16      -4.250269e-17      0.0
std    1.000006e+00      0.0  1.000006e+00       1.000006e+00      0.0
min   -2.621751e+00      0.0 -3.056169e+00      -1.344990e+00      0.0
25%   -8.922893e-01      0.0 -1.254327e+00      -3.144786e-01      0.0
50%   -2.755836e-02      0.0  5.475152e-01      -3.144786e-01      0.0
75%    8.371726e-01      0.0  5.475152e-01       7.160326e-01      0.0
max    1.121394e+01      0.0  3.550585e+00       1.823472e+01      0.0

       area_firstfloor_finished  ...    build_year     num_story  \
count              9.027500e+04  ...  9.027500e+04  9.027500e+04
mean               5.694574e-17  ...  9.970817e-16 -1.748907e-16
std                1.000006e+00  ...  1.000006e+00  1.000006e+00
min               -3.563066e+00  ... -3.520444e+00 -3.148669e-01
25%               -1.766592e-01  ... -6.507304e-01 -3.148669e-01
50%               -1.766592e-01  ...  2.449637e-02 -3.148669e-01
75%               -1.766592e-01  ...  7.841265e-01 -3.148669e-01
max                2.964825e+01  ...  1.965773e+00  9.091027e+00

       tax_building     tax_total  tax_year      tax_land  tax_property  \
count  9.027500e+04  9.027500e+04   90275.0  9.027500e+04  9.027500e+04
mean   1.550561e-17 -4.718586e-17       0.0 -6.513144e-18 -1.930232e-16
std    1.000006e+00  1.000006e+00       0.0  1.000006e+00  1.000006e+00
min   -8.606183e-01 -8.247743e-01       0.0 -6.949215e-01 -8.677957e-01
25%   -4.707308e-01 -4.661345e-01       0.0 -4.896602e-01 -4.549413e-01
50%   -2.311313e-01 -2.068912e-01       0.0 -2.131690e-01 -2.107475e-01
75%    1.450821e-01  1.494338e-01       0.0  1.675003e-01  1.340778e-01
max    4.679389e+01  4.918615e+01       0.0  6.047979e+01  4.620060e+01

       tax_delinquency_year  censustractandblock      logerror
count          9.027500e+04         9.027500e+04  9.027500e+04
mean          -6.757732e-16         4.736650e-14 -7.516680e-18
std            1.000006e+00         1.000006e+00  1.000006e+00
min           -2.045458e+01        -5.864722e-01 -2.865977e+01
25%            3.020811e-02        -5.718235e-01 -2.281952e-01
50%            3.020811e-02        -5.610437e-01 -3.387937e-02
75%            3.020811e-02         4.881393e-01  1.722320e-01
max            2.176811e+02         3.033308e+00  2.933699e+01
```

Après application du code, on remarque que les variables ont un écart-type qui est égale à un.

**Détermination du nombre des axes principaux** La deuxième étape consiste à calculer les valeurs propres de la matrice de corrélation, La fonction PCA de la classe decomposition de la librairie sklearn sur Python fait ce calcul. On peut accéder à ces valeurs avec l'attribut suivant :

**pca.explained_variance_ratio_**

```
variance expliquée:
[7.39964948 5.7401271  2.94522339 2.19039121 1.9870038  1.58269494
 1.28940268 1.1579138  1.05532472 1.0331848  1.00648635 1.00068814
 1.0001975  0.99985392 0.9978238  0.9893985  0.98417918 0.96996801
 0.94684986 0.90307072 0.84014942 0.79146738 0.74348514 0.72816178
 0.69044025 0.61626513 0.59612441 0.49603275 0.44809932]
```

La première valeur propre obtenue explique la quasi-totalité de la variance. La décroissance des valeurs propres est ici très rapide, le premier axe factoriel explique 95 % de la variance.

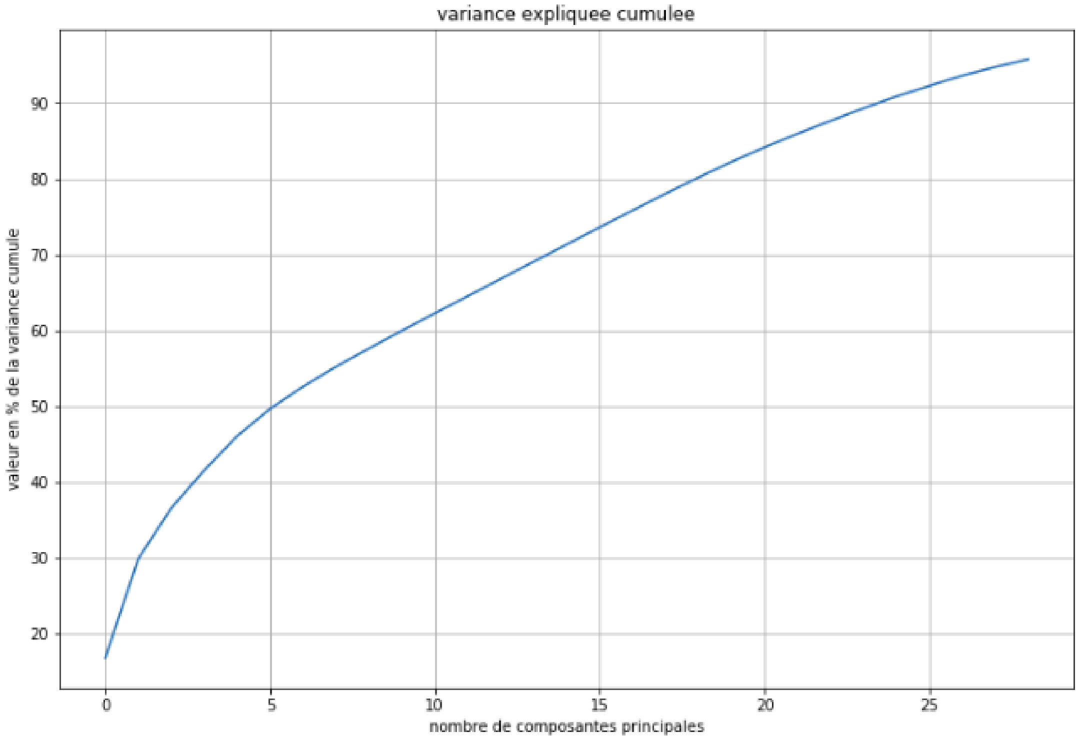
Les rapports entre valeurs propres correspondent aux carrés des rapports entre les éléments de la diagonale de la matrice de déformation. En effet, les valeurs propres sont des variances alors que les éléments de la diagonale de la matrice de déformation ont ici une signification d'écart-types.

Le pourcentage de la variable expliquée par chaque variable est donnée par le code suivant :

**pca.explained_variance_ratio_**

```
pourcentage de la variance expliquée
[0.16817168 0.29862743 0.36556346 0.41534443 0.46050303 0.4964729
 0.52577713 0.55209301 0.57607735 0.59955852 0.62243292 0.64517553
 0.667907   0.69063066 0.71330818 0.73579422 0.75816164 0.78020608
 0.80172512 0.82224919 0.84134325 0.85933091 0.87622808 0.892777
 0.90846862 0.92247446 0.93602257 0.9472959  0.95747984]
```

Le graphe suivant présente le pourcentage de la varaince expliquée cumulé des variables


variance expliquee cumulee
valeur en % de la variance cumule
nombre de composantes principales
90
80
70
60
50
40
30
20
0
5
10
15
20
25

# Chapitre 2

# La méthode XGBoost

**Le Boosting de Gradient** est un algorithme d'apprentissage supervisé dont le principe et de combiner les résultats d'un ensemble de modèles plus simple et plus faibles afin de fournir une meilleur prédiction.

L'algorithme **XGBoost** est un algorithme ensembliste qui agrège des arbres.
À chaque itération, le nouvel arbre apprend de l'erreur commise par l'arbre précédent. Ainsi, même si chaque arbre a un pouvoir prédictif faible, la règle de décision construite en sommant le résultat de chaque arbre est elle très fiable.
Dans le cadre de notre problématique, ce genre d'algorithme a l'avantage de pouvoir utiliser toute l'information dont on dispose sur les proprietés.

## 2.1 Formalisation mathématique de XGBoost :

**Extreme Gradient Boosting (XGBoost)** , est un nouveau classificateur basé sur un ensemble d'arbres de classification et de régression.
Dans XGBoost, les arbres sont optimisés en utilisant le gradient boosting.
Soit la sortie d'un arbre :

$$f(x) = w_q(x_i)$$

où $x$ est le vecteur d'entrée et $w_q$ est le score du feuille $q$. Le résultat d'un ensemble de $K$ arbres sera :

$$y_i = \sum_{k=1}^{k} f_k(x_i)$$

L'algorithme XGBoost tente de minimiser la fonction objectif suivante $J$ au point $t$ :

$$J(t) = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i)$$

où le premier terme contient la fonction de perte de train $L$ (par exemple, moyenne erreur au carré) entre la classe réelle $y$ et la sortie $\hat{y}$ pour les $n$ échantillons et le deuxième terme est le terme de régularisation, qui contrôle la complexité du modèle et permet d'éviter le sur-apprentissage. Dans XGBoost, la complexité est définie comme suit :

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

où $T$ est le nombre de feuilles, $\gamma$ est la pseudo-régularisation hyperparamètre, en fonction de chaque jeu de données et $\lambda$ est la norme L2 pour les poids des feuilles.

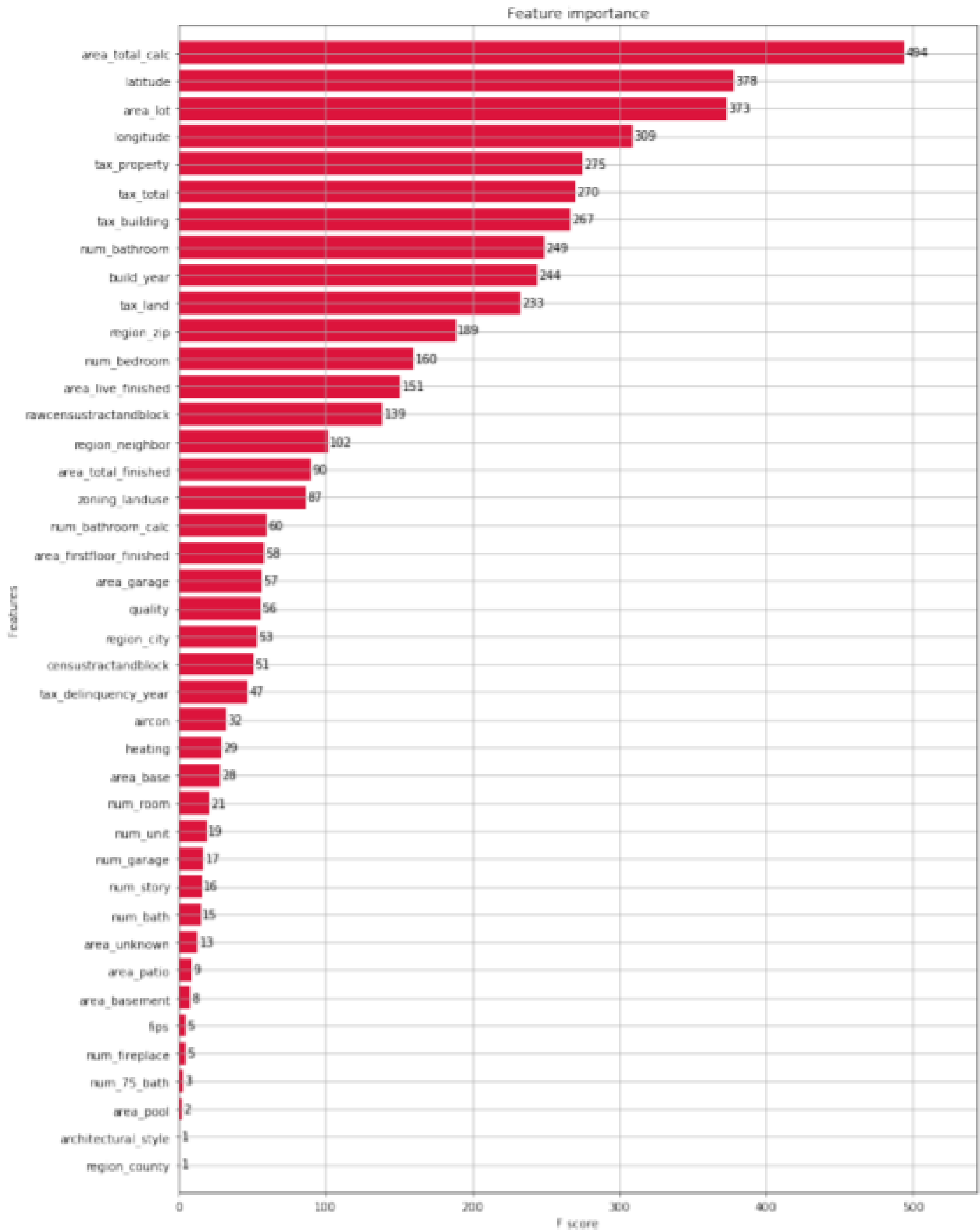
Application de gradients pour l'approximation du deuxième ordre de la fonction de perte et trouver les poids optimaux $w$, la valeur optimale de l'objectif la fonction est :

$$f(t) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I g_i})^2}{\sum_{i \in I} h_i + \lambda} + \gamma T$$

où $g_i = \delta_{\hat{y}^{t-1}} \mathrm{L}(y, \hat{y}^{t-1})$ et $h_i = \delta^2_{\hat{y}^t\_1} \mathrm{L}(y, \hat{y}^{t-1})$ sont les des statistiques de gradient sur la fonction
de perte, et $I$ est l'ensemble des feuilles.

## 2.2 Application de XGBoost sur le jeu de données en Python :

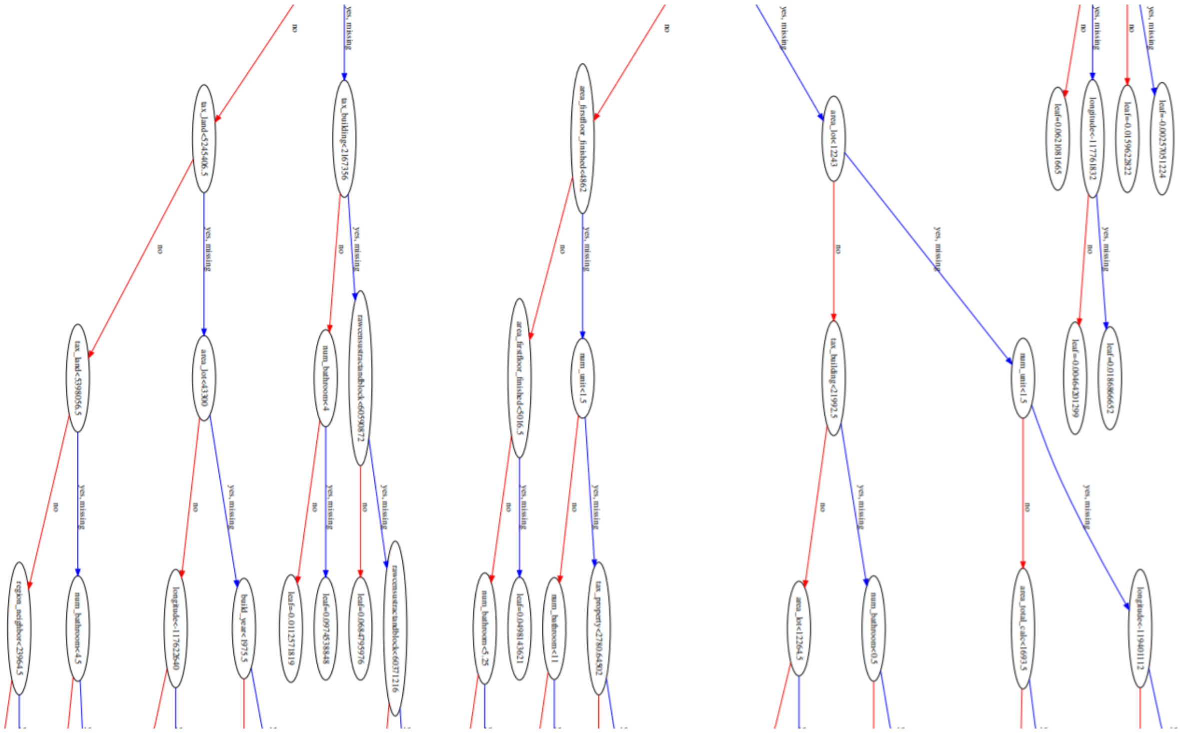
- Études de l'importance des variables : liens entre les attributs avec XGBoost




On a obtenu le classement de l'importance des caractèristiques des propriétés selon cet ordre avec la méthode xgboost.

```
xgb.to_graphviz(model, num_trees=1,rankdir='LR')
```

À l'aide de cette commande on affiche l'arbre de décision :



**Un Extrait de l'arbre de décision de XGBoost**