UNIVERSITÉ NATIONALE DU VIETNAM

INSTITUT FRANCOPHONE INTERNATIONAL

Promotion : 24

Option : SIM (systèmes Intélligents et multimédias)

Fouille de données

Analyse des données

Rédigé par :

MBIAYA KWUITE Franck Anael

KANA NGUIMFACK Kevin

Enseignant :

NGUYEN Thi Minh Huyen

Année Académique 2019/2020

SOMMAIRE

INTRODUCTION

L'ensemble des données soumit à notre étude provient de la plate-forme kaggle.com. Il comporte une description détaillée des maisons provenant des sociétés immobilières du comté de King, dans l'État de Washington, aux États-Unis. L'ensemble de données se compose de données historiques sur les maisons vendues entre mai 2014 et mai 2015. L'ensemble de données comprend 21 variables et 21613 observations.

Notre objectif dans le cadre de ce rapport consistera dans un premier temps à faire une analyse descriptive et exploratoire de ces données, Ensuite nous appliquerons des méthodes d'apprentissage non supervisé et plus particulierement une analyse factorielle et un clustering. Nous finirons par un apprentissage supervisé en utilisant plusieurs algorithmes existants en nous comparerons les résultats obtenus.

CHAPITRE 1

ANALYSE DES DONNÉES

Après analyse de ces données, on se rend compte que l'on a 14 variables de type entier, 6 variables de type réel et une variable de type date. La repartition des types de variable est présentée dans la figure suivante.

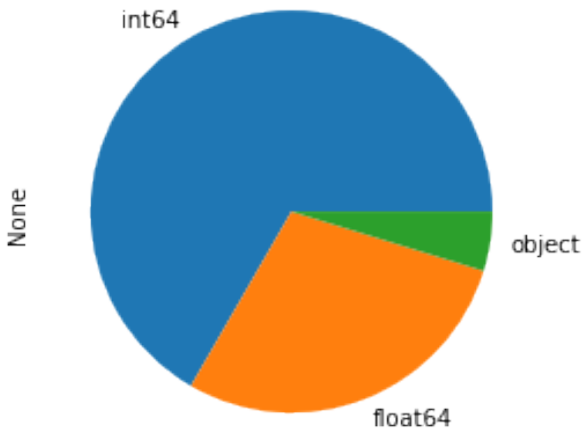


FIGURE 1.1 – Repartition des types de variables dans le jeu de données

La description des 21 variables de notre jeu de données est la suvante :

— **id :** C'est un identifiant d'une entrée du jeu de données. Il est représetée sous format entier.
— **date :** C'est la date de la mise en vente des différentes maison. Elle est représentée sous forme de texte comprenant l'année, le mois et le jour.
— **price :** C'est le prix de la maison. Elle est présentée sous format réel
— **bedrooms :** C'est le nombre de chambre dans une maison. Elle est présentée sous format entier
— **bathrooms :** C'est le nombre de salle de bain dans une maison. Elle est présentée sous format réel
— **sqft_living :** C'est la superficie de la maison. Elle est présentée sous format entier
— **sqft_lot :** C'est la superficie du lot sur lequel se trouve la maison. Elle est présentée sous format entier
— **floors :** C'est le nombre d'étage de la maison. Elle est présentée sous format réel
— **waterfront :** Elle définie si la maison est avec vue sur la mer. Elle est présentée sous format entier
— **view :** Elle définie le nombre de fois que la maison à été vue. Elle est présentée sous format entier
— **condition :** Elle définie le niveau de conditionnement de la maison. Elle est présentée sous format entier

— **grade :** Elle représente la note globale attribuée à la maison, basé sur le système de classement du compté de King. Elle est présentée sous format entier
— **sqft_above :** Elle représente la superficie de la maison en dehors du sous-sol. Elle est présentée sous format réel
— **sqft_basement :** Elle représente la superficie du sous-sol. Elle est présentée sous format entier
— **yr_built :** C'est l'année de construction de ma maison. Elle est présentée sous format entier
— **yr_renovated :** C'est l'année de rénovation de la maison. Elle est présentée sous format entier
— **zipcode :** C'est le code postal de la maison. Elle est présentée sous format entier
— **lat :** C'est la latitude de la maison. Elle est présentée sous format réel
— **long :** C'est la longitude de la maison. Elle est présentée sous format réel
— **sqft_living15 :** C'est la superficie de la maison réelle en 2015. Elle est présentée sous format entier
— **sqft_lot15 :** c'est la superficie du lot sur lequel se trouve la maison en 2015. Elle est présentée sous format entier

1.1 Analyse descriptive

Pour la suite de notre analyse, nous allons supprimer les variables **id**, **date** et **zipcode** qui ne sont pas importantes. Nous avons alors 18 variables à analyser. Notre analyse portera donc sur les données ayant les caractéristiques suivantes :

— **Nombre de variable :** 18
— **Nombre d'observations :** 21613
— **Types de variable :** Entier et Réels
— **Valeur manquante :** 2 valeurs manquantes dans la variable sqft_above
— **Variable label :** Price

1.1.1 La variable price

la variable price représente le prix des différentes maisons.

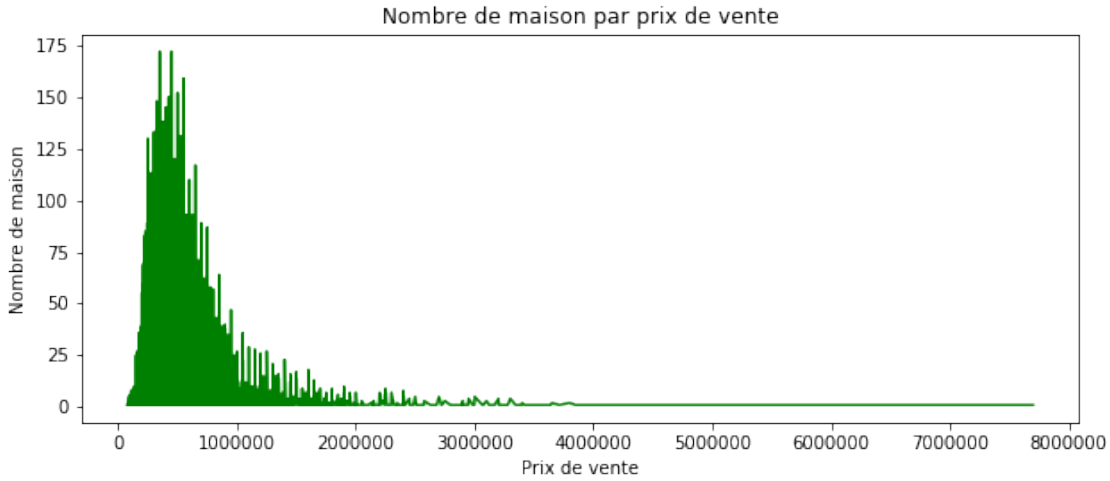


FIGURE 1.2 – Nombre de maison par prix de vente

Tendances centrales

- **Max** : Le prix maximal d'une maison est **7700000**

- **Min** : Le prix minimal d'une maison est **75000**
- **Mode** : Le prix le plus fréquent pour une maison est **350000**
- **Médiane** : Le prix pour lequel 50% des maisons sont en dessous est **450000**
- **Moyenne** : Le prix moyen des maisons est **540088,14**

Dispertion

- **Variance** : **134776142225.6**
- **Ecart type** : **367118,70**

Interprétation

L'écart type représente plus de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable price sont dispersées.

1.1.2 La variable bedrooms

bedrooms représente le nombre de chambre présente dans chaque maison.

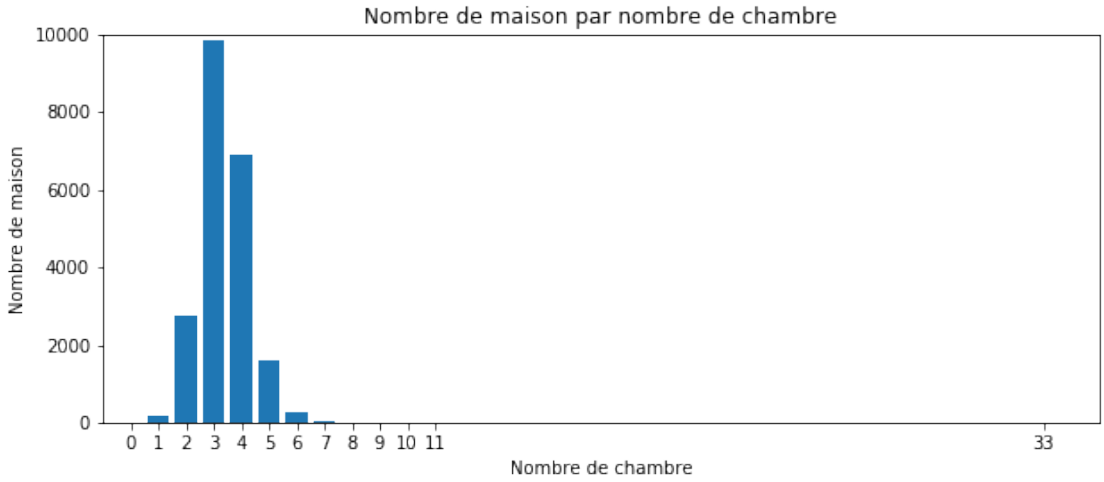


FIGURE 1.3 – Nombre de chambre par maison

Tendances centrales

- **Max** : Le nombre de chambre maximal dans une maison est **33**
- **Min** : Le nombre minimal de chambre dans une maison est **0**
- **Mode** : Le nombre de chambre le plus fréquent dans une maison est **3**
- **Médiane** : Le nombre de chambre pour laquelle 50% des maisons sont en dessous est **3**
- **Moyenne** : Le nombre de chambre moyen par maison est **3,37**

Dispertion

- **Variance** : **0,86**
- **Ecart type** : **0,93**

Interprétation

L'écart type représente moins de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable bedrooms ne sont pas trop dispersées.

1.1.3 La variable bathrooms

bathrooms représente le nombre de salle de bain dans chaque maison.

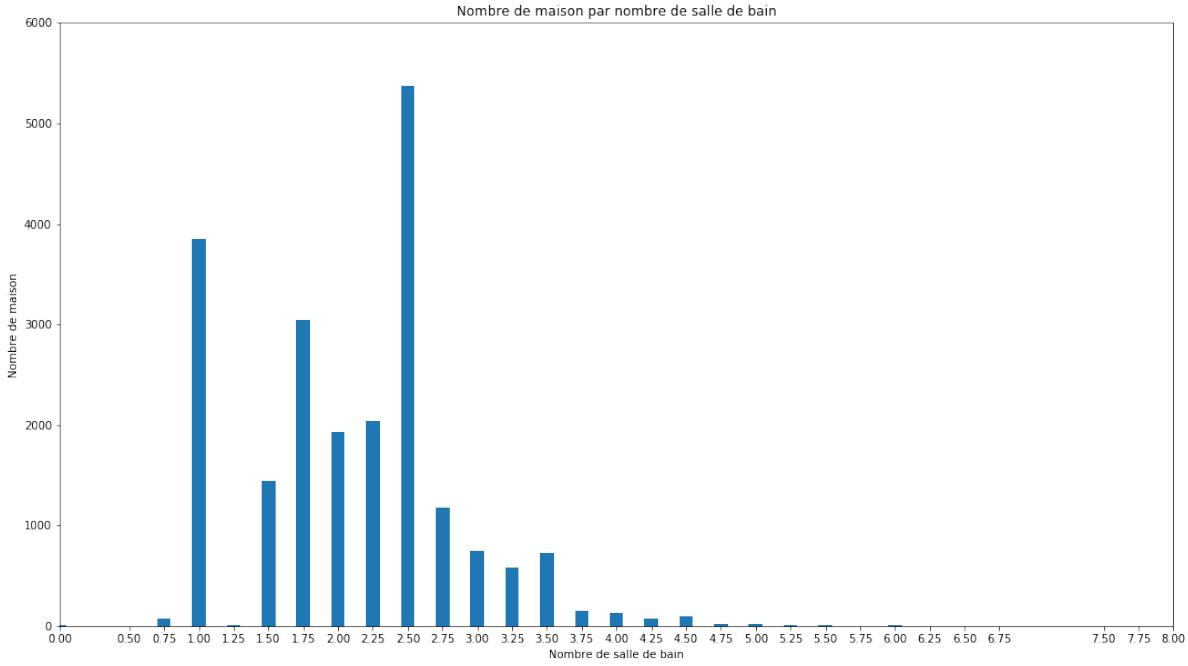


FIGURE 1.4 – Nombre de salle de bain par maison

Tendances centrales

- **Max** : Le nombre de salle de bain maximal dans une maison est **8**
- **Min** : Le nombre minimal de salle de bain dans une maison est **0**
- **Mode** : Le nombre de salle de bain le plus fréquent dans une maison est **2,5**
- **Médiane** : Le nombre de salle de bain pour laquelle 50% des maisons sont en dessous est **2.25**
- **Moyenne** : Le nombre de salle de bain moyen par maison est **2,12**

Dispertion

- **Variance** : **0,59**
- **Ecart type** : **0,77**

Interprétation

L'écart type représente moins de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable bathrooms ne sont pas trop dispersées.

1.1.4 La variable sqft_living

sqft_living représente le superficie de chaque maison.

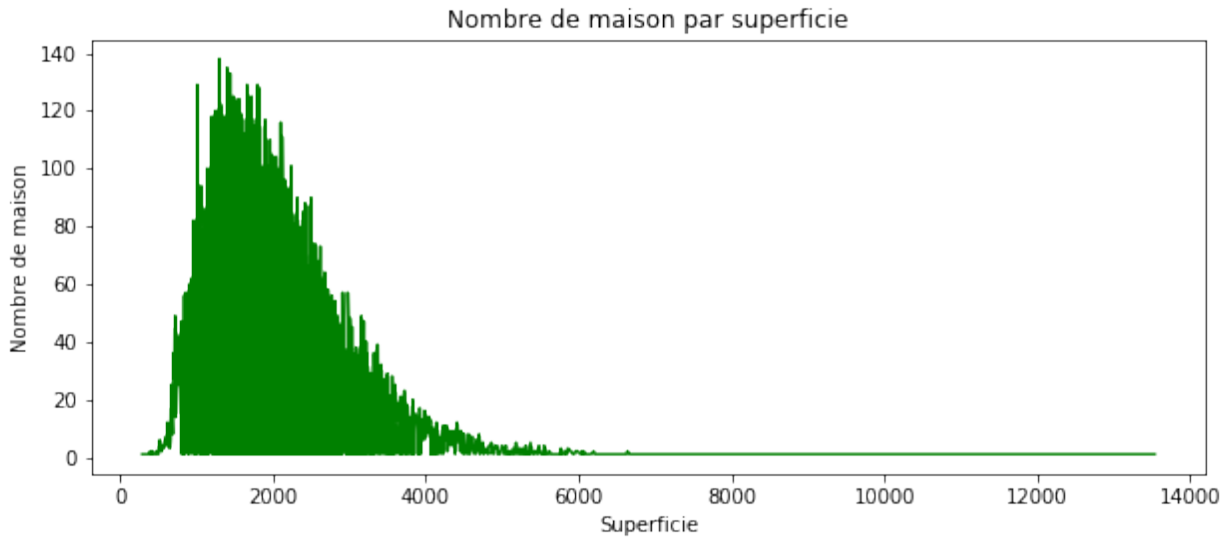


FIGURE 1.5 – Nombre de maison en fonction de la superficie

Tendances centrales

- **Max** : La superficie maximale d'une maison est **13540**
- **Min** : La superficie minimale d'une maison est **290**
- **Mode** : La superficie la plus fréquente des maisons est **1300**
- **Médiane** : La superficie pour laquelle 50% des maisons sont en dessous est **1910**
- **Moyenne** : La superficie moyenne des maisons est **2079,9**

Dispertion

- **Variance** : **843494,65**
- **Ecart type** : **918,42**

Interprétation

L'écart type représente moins de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable sqft_living ne sont pas trop dispersées.

1.1.5 La variable sqft_lot

sqft_lot représente la superficie du terrain sur le quel se trouve chaque maison.

FIGURE 1.6 – Nombre de maison par superficie du lot sur lequel se trouve la maison

Tendances centrales

- **Max** : La superficie maximale du terrain d'une maison est **1651359**
- **Min** : La superficie minimale du terrain d'une maison est **520**
- **Mode** : La superficie la plus fréquente d'une maison est **5000**
- **Médiane** : La superficie de terrain pour laquelle 50% des maisons sont en dessous est **7618**
- **Moyenne** : La superficie moyenne de terrain des maisons est **15106,97**

Dispertion

- **Variance** : **1715658774,18**
- **Ecart type** : **41420,51**

Interprétation

L'écart type représente bien plus de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable sqft_lot sont très dispersées.

1.1.6 La variable floors

floors représente le nombre d'étage de chaque maison.

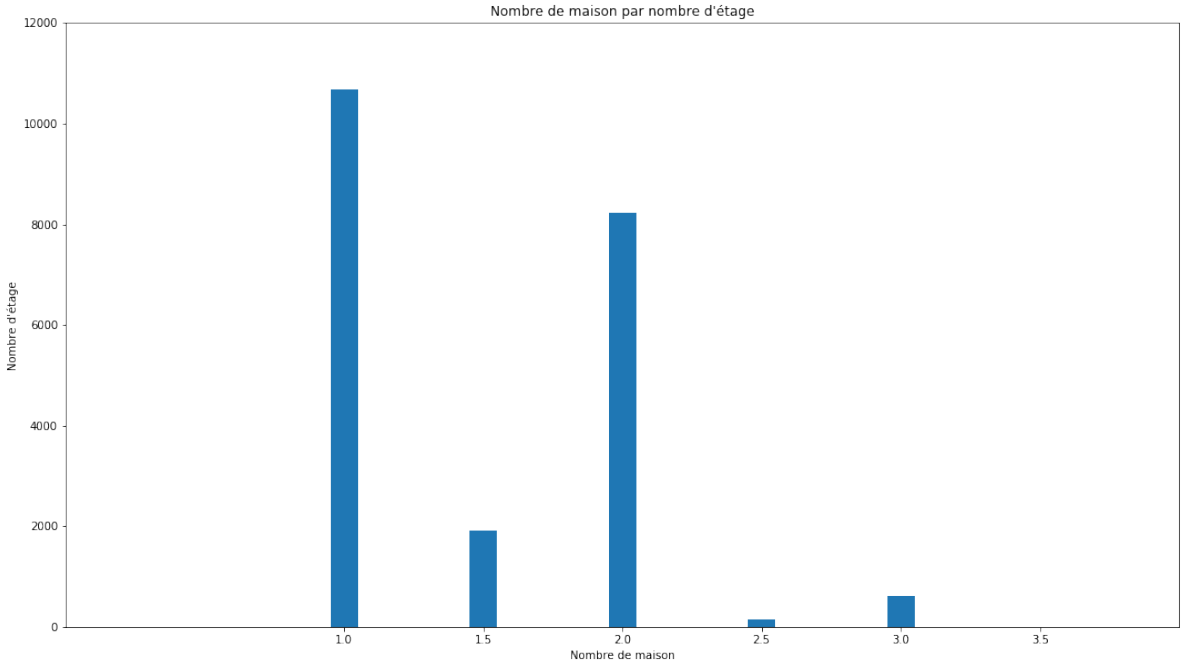


FIGURE 1.7 – Nombre de maison par nombre d'étage

Tendances centrales

- **Max** : Le nombre maximal d'étage d'une maison est **3.5**
- **Min** : Le nombre minimale d'étage d'une maison est **1**
- **Mode** : Le nombre d'étage le plus fréquent d'une maison est **1**
- **Médiane** : Le nombre d'étage pour lequel 50% des maisons sont en dessous est **1,5**
- **Moyenne** : Le nombre d'étage moyen des maisons est **1,5**

Dispertion

- **Variance** : **0,29**
- **Ecart type** : **0,54**

Interprétation

L'écart type représente moins de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable floors ne sont pas trop dispersées.

1.1.7 La variable waterfront

waterfront permet de dire si une maison est avec vue sur la mer ou non. Elle est représentée sous forme numérique ou 0 signifie Non et 1 signifie Oui.

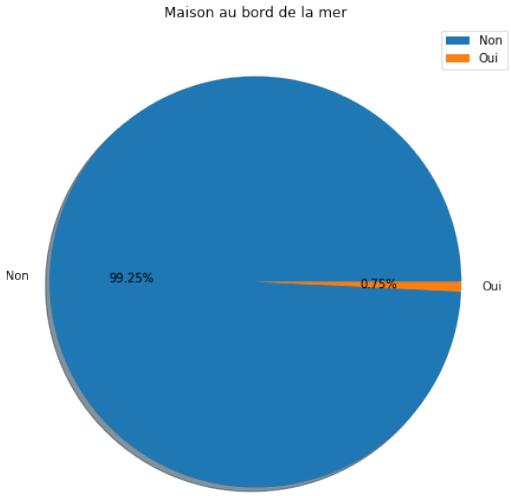


FIGURE 1.8 – Pourcentage des maisons au bond de la mer

1.1.8 La variable view

La variable view représente le nombre de fois que la maison à été vue. Elle est présenté sous format numérique.

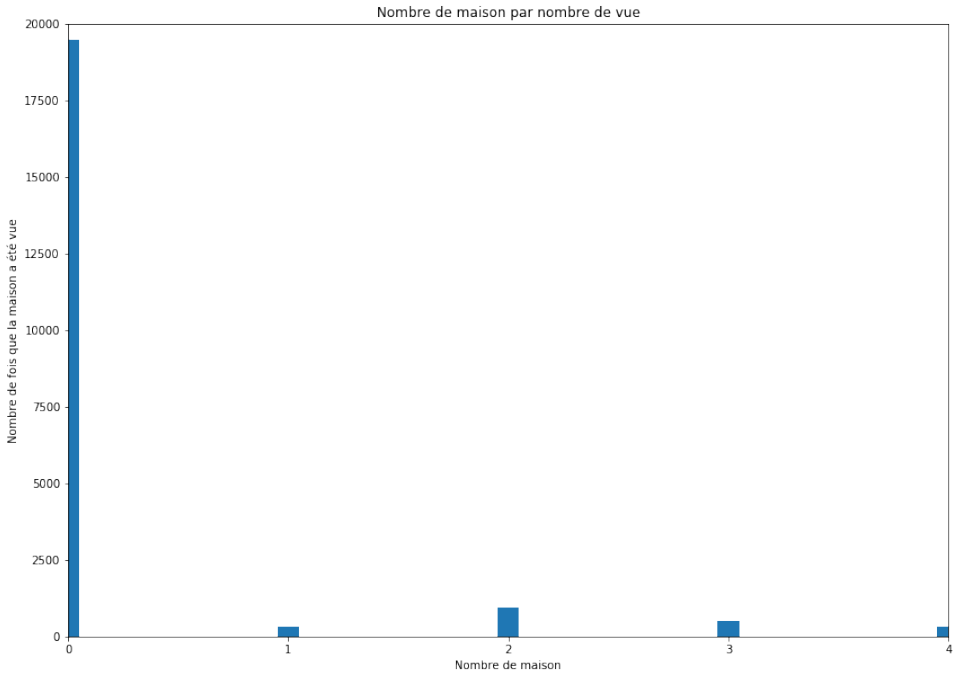


FIGURE 1.9 – Nombre de maison par nombre de vue

Tendances centrales

- **Max** : Le nombre maximal de vue d'une maison est **4**
- **Min** : Le nombre minimale de vue d'une maison est **0**
- **Mode** : Le nombre de vue le plus fréquent d'une maison est **0**
- **Médiane** : Le nombre de vue pour lequel 50% des maisons sont en dessous est **0**
- **Moyenne** : Le nombre de vue moyen des maisons est **0,23**

Dispertion

- **Variance** : **0,59**
- **Ecart type** : **0,77**

Interprétation

L'écart type représente plus de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable view sont dispersées.

1.1.9 La variable condition

condition permet de décrire la condition de la maison. Elle est représentée sous forme numérique allant de 0 à 5 en fonction de la condition de la maison.

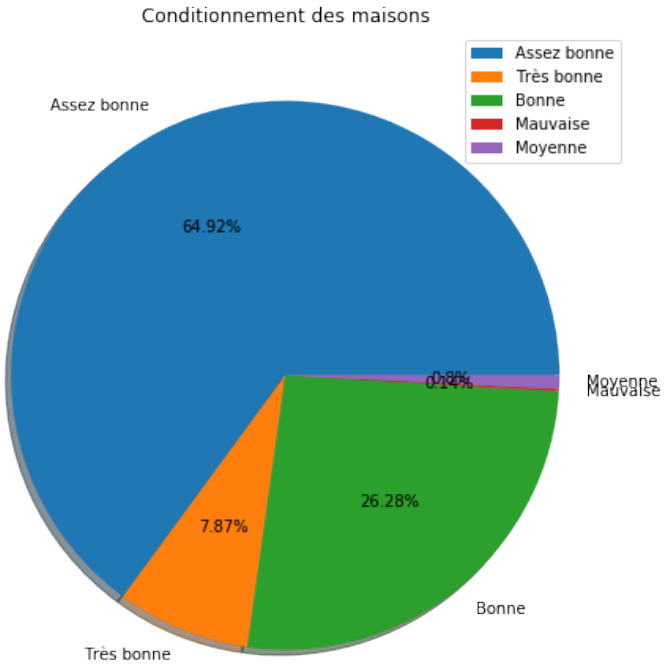


FIGURE 1.10 – Condition des maisons

1.1.10 La variable grade

grade représente la note globale attribuée à la maison, basé sur le système de classement du compté de King. Elle est présenté sous format numérique compris entre 0 et 13.

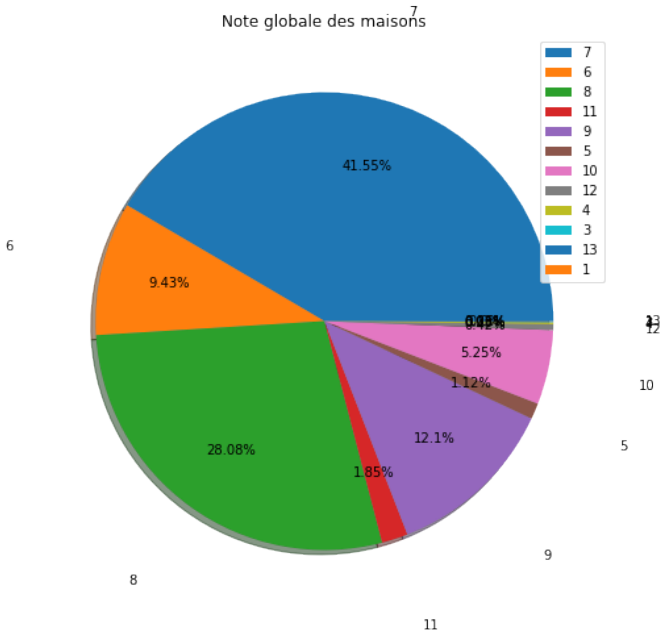


FIGURE 1.11 – Note globale des maisons

1.1.11 La variable sqft_above

La variable sqft_above représente la superficie de la maison en dehors du sous-sol. Cette variable contient 2 valeurs manquantes que nous allons remplacer par la valeur moyenne de la variable.

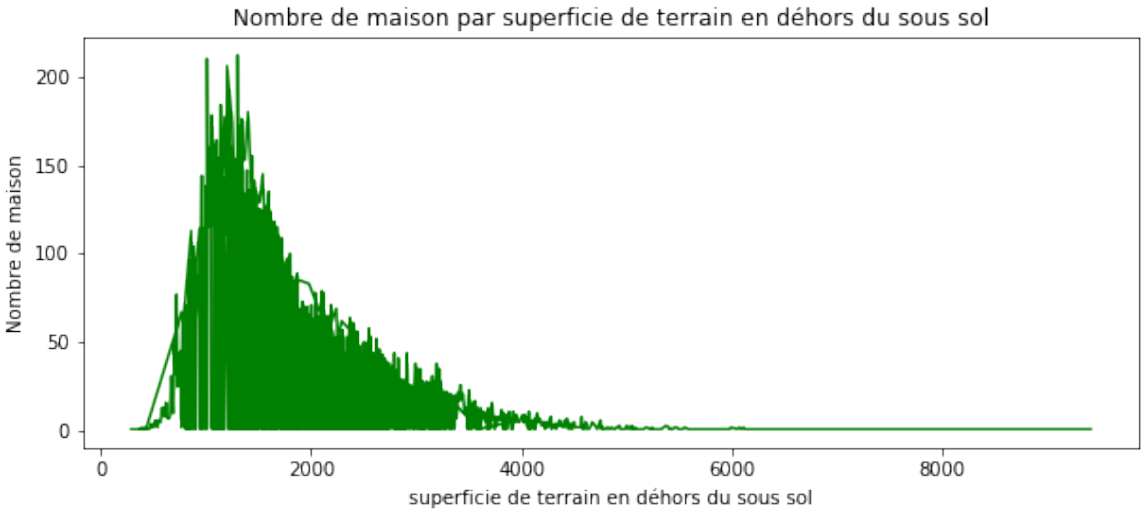


FIGURE 1.12 – Nombre de maison par superficie en dehors du sous sol

Tendances centrales

- **Max** : Le nombre maximal de la superficie d'une maison en dehors du sous sol est **9410**
- **Min** : Le nombre minimale de la superficie d'une maison en dehors du sous sol est **290**
- **Mode** : La superficie la plus fréquente est **1300**
- **Médiane** : La superficie pour lequel 50% des maisons sont en dessous est **1560**
- **Moyenne** : La superficie moyenne des maisons en dehors du sous sol est **1788,4**

Dispertion

- **Variance** : **685701,06**
- **Ecart type** : **828,07**

Interprétation

L'écart type représente plus de la motié de la moyenne, nous pouvons donc conclure que les données représentant la variable sqft_above ne sont pas dispersées.

1.1.12 La variable sqft_basement

La variable sqft_basement représente la superficie du sous-sol.

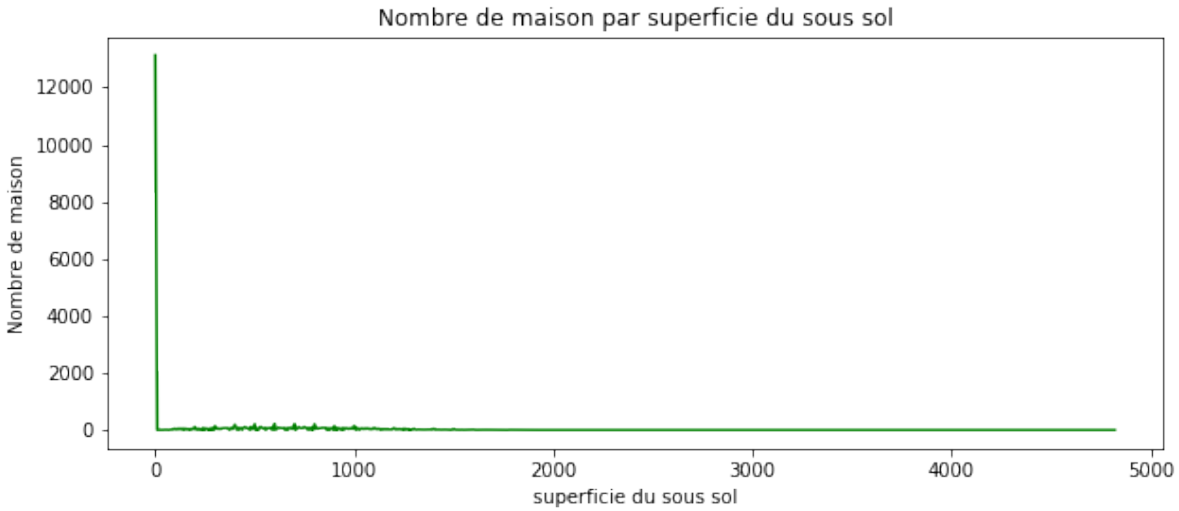


FIGURE 1.13 – Nombre de maison par superficie du sous sol

Tendances centrales

- **Max** : La valeur maximal de la superficie du sous sol est **4820**
- **Min** : La valeur minimale de la superficie du sous sol est **0**
- **Mode** : La superficie du sous sol la plus fréquente est **0**
- **Médiane** : La superficie du sous sol pour laquelle 50% des maisons sont en dessous est **0**
- **Moyenne** : La superficie moyenne du sous sol est **291,5**

Dispertion

- **Variance** : **195863,6**
- **Ecart type** : **442,56**

Interprétation

L'écart type représente le double de la moyenne, nous pouvons donc conclure que les données représentant la variable sqft_basement sont très dispersées.

1.1.13 La variable yr_built

La variable yr_built représente la date de construction des maison.

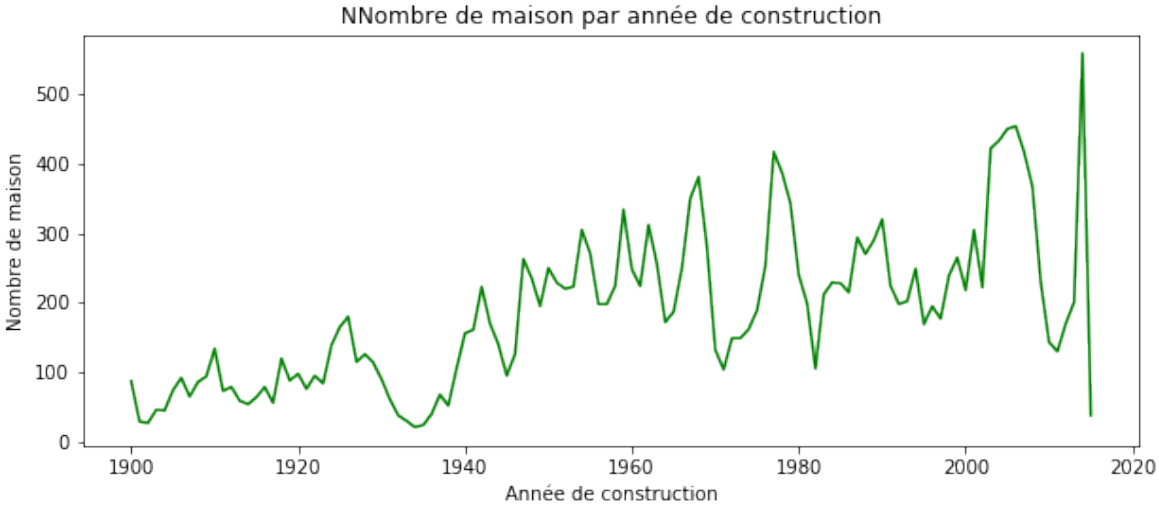


FIGURE 1.14 – Nombre de maison par année de construction

Tendances centrales

- **Max** : L'année de construction la plus récente est **2015**
- **Min** : L'année de construction la plus ancienne est **1900**
- **Mode** : L'année ayant le plus de construction est **2014**
- **Médiane** : l'année de construction pour laquelle 50% des maisons sont en dessous est **1975**
- **Moyenne** : L'année de construction moyenne est **1971**

1.1.14 La variable yr_renovated

La variable yr_renovated représente la date de renovation des maisons.

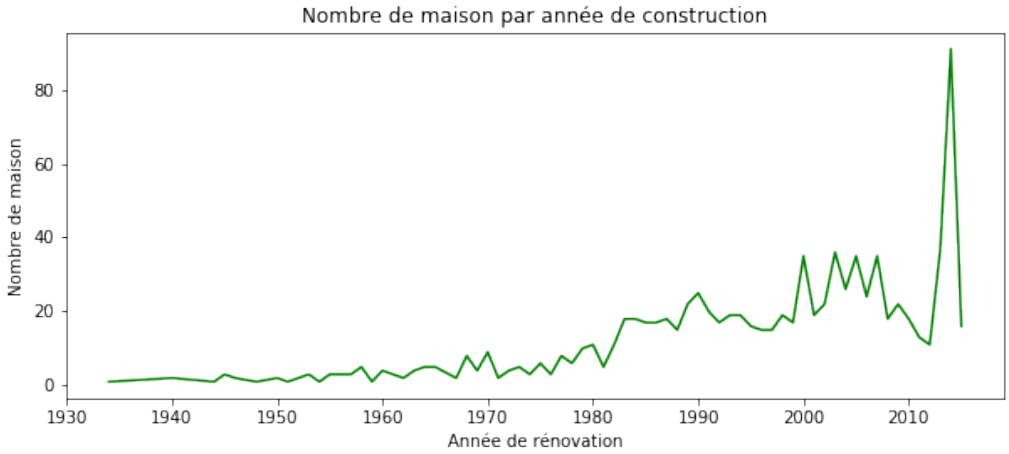


FIGURE 1.15 – Nombre de maison par année de renovation

Tendances centrales

- **Max** : L'année de renovation la plus récente est **2015**
- **Min** : L'année de renovation la plus ancienne est **1934**

1.1.15 La variable lat

La variable lat représente la latitude de la maison.

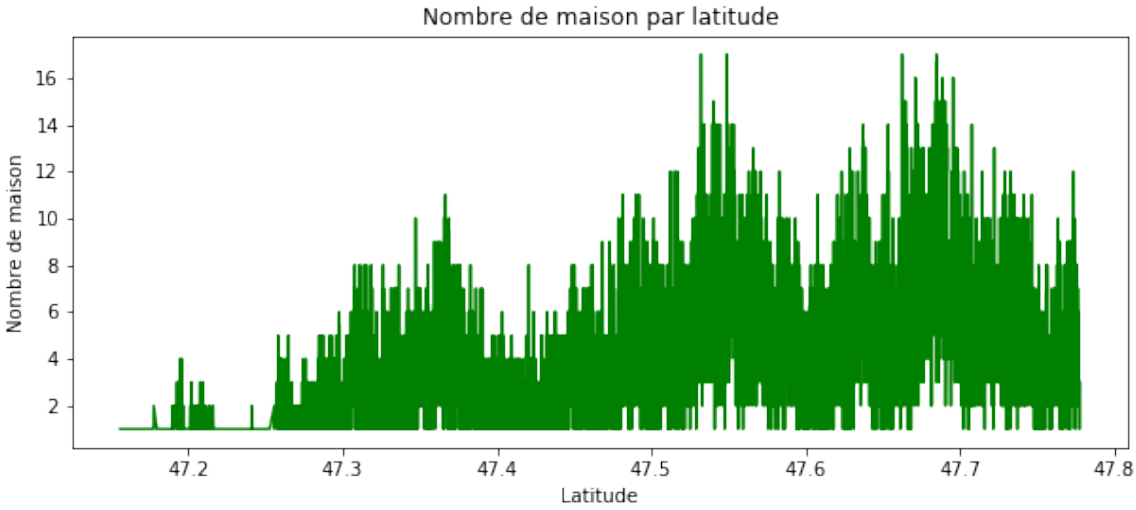


FIGURE 1.16 – Nombre de maison par latitude

Tendances centrales

- **Max** : La valeur maximal de la latitude est **47,7776**
- **Min** : La valeur minimale de la latitude est **47,1559**
- **Mode** : La latitude la plus fréquente est **47,5322**
- **Médiane** : La latitude pour laquelle 50% des maisons sont en dessous est **47,57**
- **Moyenne** : La latitude moyenne est **47,56**

Dispertion

- **Variance** : **0,019**
- **Ecart type** : **0,138**

Interprétation

Nous pouvons conclure que les données représentant la variable lat ne sont pas dispersées.

1.1.16 La variable long

La variable long représente la longitude de la maison.

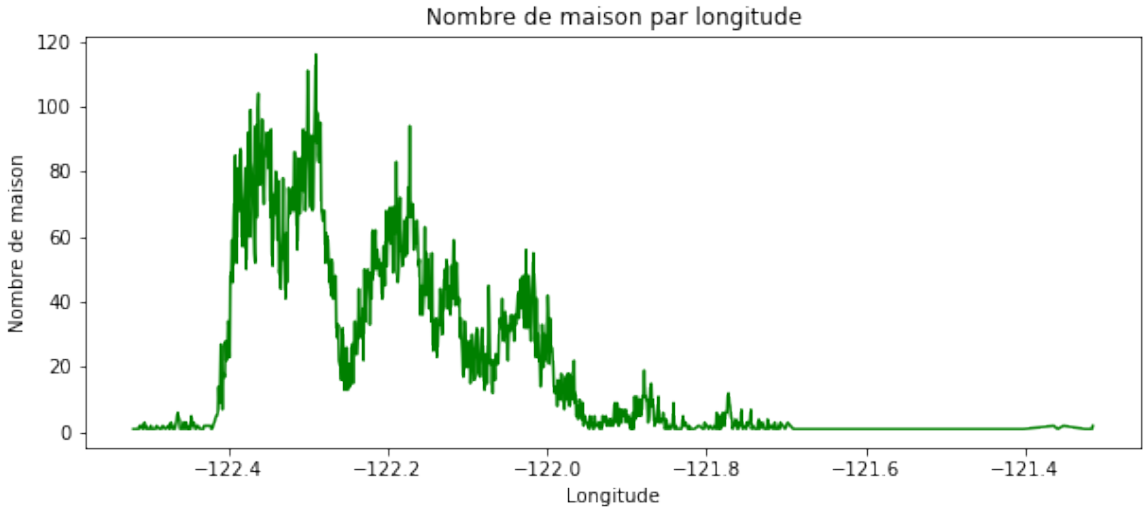


FIGURE 1.17 – Nombre de maison par longitude

Tendances centrales

- **Max** : La valeur maximal de la longitude est **-121,315**
- **Min** : La valeur minimale de la longitude est **-122,518**
- **Mode** : La longitude la plus fréquente est **-122,29**
- **Médiane** : La longitude pour laquelle 50% des maisons sont en dessous est **-122,23**
- **Moyenne** : La longitude moyenne est **-122,21**

Dispertion

- **Variance** : **0,0198**
- **Ecart type** : **0,14**

Interprétation

Nous pouvons conclure que les données représentant la variable long ne sont pas dispersées.

1.1.17 La variable sqft_living15

La variable sqft_living15 représente la superficie réelle des différentes maisons en 2015.

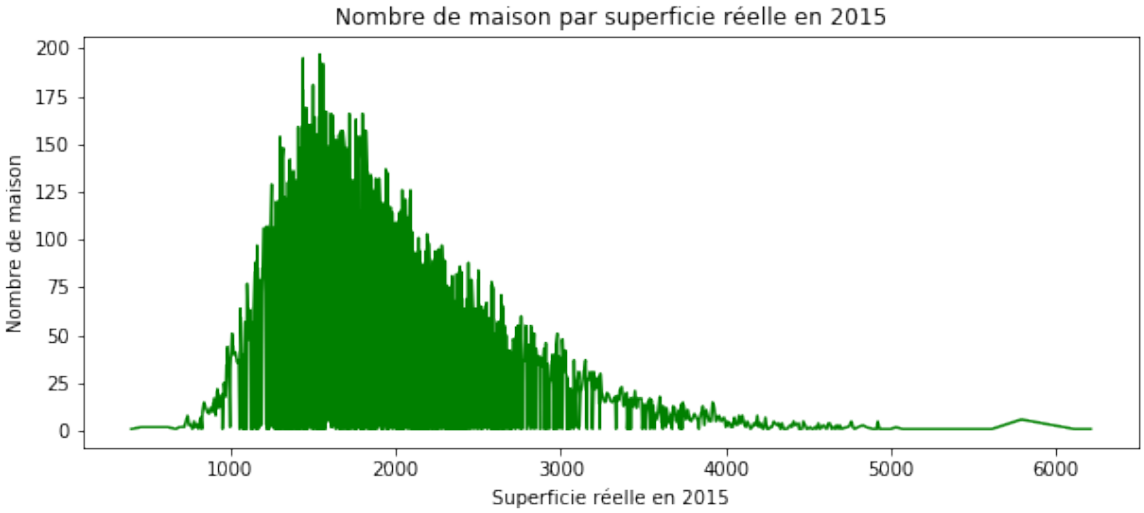


FIGURE 1.18 – Nombre de maison par superficie réelle en 2015

Tendances centrales

- **Max** : La valeur maximal de la superficie réelle en 2015 est **6210**
- **Min** : La valeur minimale de la la superficie réelle en 2015 est **399**
- **Mode** : La la superficie réelle en 2015 la plus fréquente est **1540**
- **Médiane** : La la superficie réelle en 2015 pour laquelle 50% des maisons sont en dessous est **1840**
- **Moyenne** : La la superficie réelle en 2015 moyenne est **1986,55**

Dispertion

- **Variance** : **469739,50**
- **Ecart type** : **685,37**

Interprétation

Nous pouvons conclure que les données représentant la variable sqft_living15 ne sont pas dispersées.

1.1.18 La variable sqft_lot15

La variable sqft_lot15 représente la superficie réelle du lot des différentes maisons en 2015.

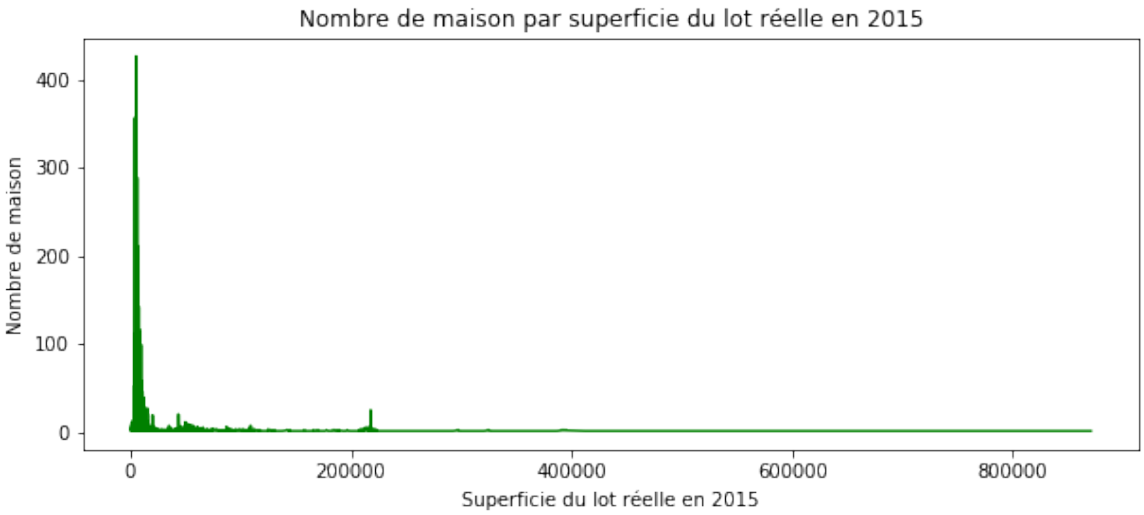


FIGURE 1.19 – Nombre de maison par superficie du lot réelle en 2015

Tendances centrales

- **Max** : La valeur maximal de la superficie du lot réelle en 2015 est **871200**
- **Min** : La valeur minimale de la la superficie du lot réelle en 2015 est **651**
- **Mode** : La la superficie du lot réelle en 2015 la plus fréquente est **5000**
- **Médiane** : La la superficie du lot réelle en 2015 pour laquelle 50% des maisons sont en dessous est **1840**
- **Moyenne** : La la superficie du lot réelle en 2015 moyenne est **12768,455**

Dispertion

- **Variance** : **745483731,36**
- **Ecart type** : **27303,54**

Interprétation

Nous pouvons conclure que les données représentant la variable sqft_living15 sont très dispersées.

1.2 Analyse exploratoire

L'analyse exploratoire consiste à analyser les relations qui existent entre les différentes differentes variables du jeu de donées. Pour cela, nous nous interesserons à la correlation qui existe entre les variables. Nous utiliserons ici à la matrice de correlation suivantes :

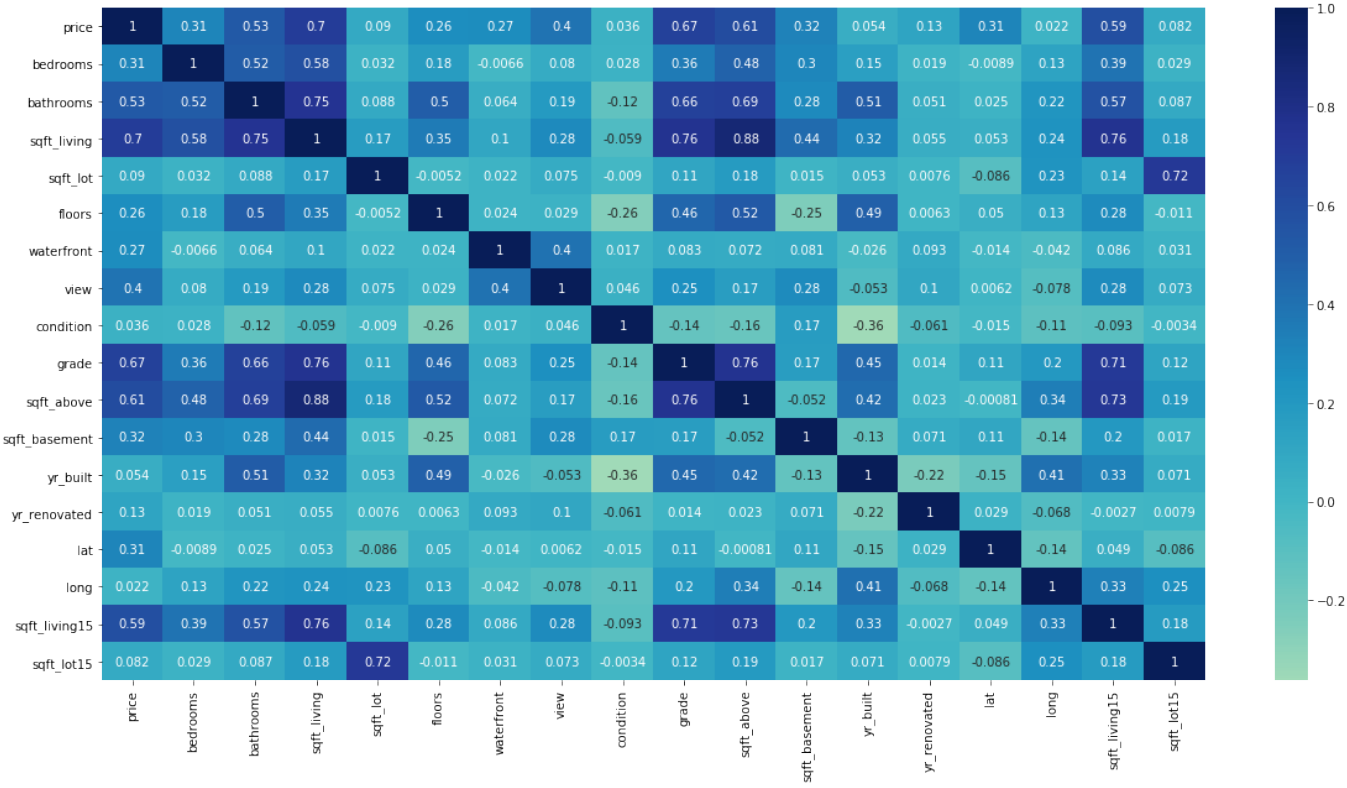


FIGURE 1.20 – Relations entre variables

Interprétation : Nous remarquons que les variables **sqft_living**, **bathrooms**, **grade**, **sqft_above**, et **sqft_above** sont correlées.

La variable **sqft_lot** est assez correlée avec la variable **sqft_lot15**. également, la variable **price** est correlée avec les variables **sqft_living**, et **grade** comme le montre la figure suivante.

```
long           0.021626
condition      0.036362
yr_built       0.054012
sqft_lot15     0.082447
sqft_lot       0.089661
yr_renovated   0.126434
floors         0.256794
waterfront     0.266369
lat            0.307003
bedrooms       0.308350
sqft_basement  0.323816
view           0.397293
bathrooms      0.525138
sqft_living15  0.585379
sqft_above     0.605567
grade          0.667434
sqft_living    0.702035
price          1.000000
Name: price, dtype: float64
```

FIGURE 1.21 – correlation avec la variable price

CHAPITRE 2

CLASSIFICATION DES DONNÉES

2.1 Analyse factorielle

Comme nous n'avons que des variables quantitatives, notre analyse factorielle consistera à faire une **Analyse en Composantes Principales (ACP)**. Pour cela, nous allons commencer par centrer et réduire les données pour qu'ils soient tous dans la même unité de mésure afin de pouvoir les comparer.

2.1.1 Analyse des corrélations entre les variables

L'analyse des corrélations entre les variables consiste à étudier les liaisons qui existent entre les variables, et au besoin, les regrouper en de nouvelles variables synthétiques. Pour cela nous utiliserons le cercle de corrélations. Il nous fait à présent déterminer le nombre de variables synthétiques que nous allons utiliser. Pour cela, nous allons choisir le nombre d'axes qui permet de décrire au mieux les variables de notre jeu de données.

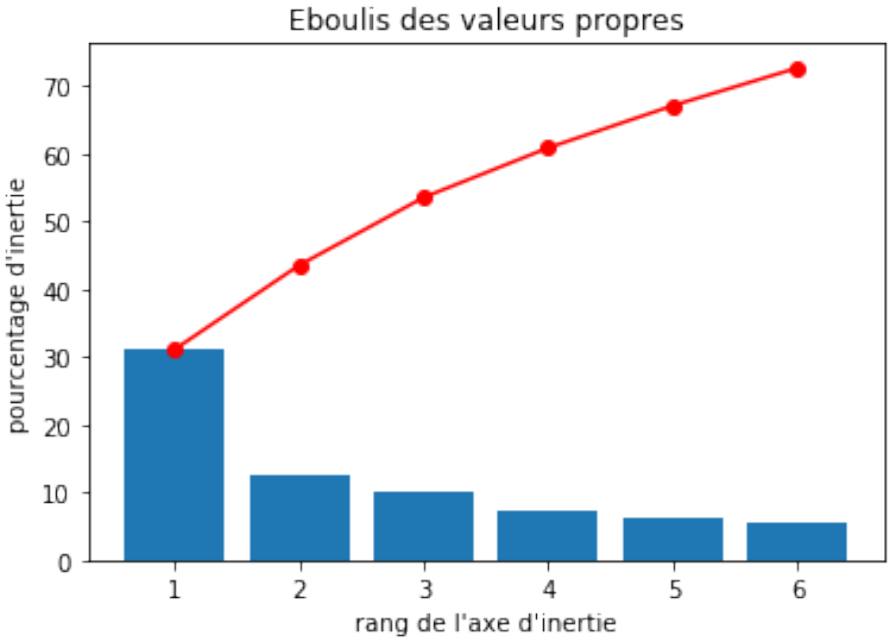


FIGURE 2.1 – Analyse de la représentativité des variables synthétiques

Nous pouvons donc utiliser 6 variables synthétiques, ce qui nous permet de décrire environs 71% de nos variables selon la distribution suivante :

- Première variable : 31,0%
- Deuxième variable : 12,4%
- Troisième variable : 10,0%
- Quatrième variable : 7,3%
- Cinquième variable : 6,2%
- Sixième variable : 5,6%

Variables synthétiques F1 et F2

Le cercle des corrélations des variables F1 et F2 est le suivant :

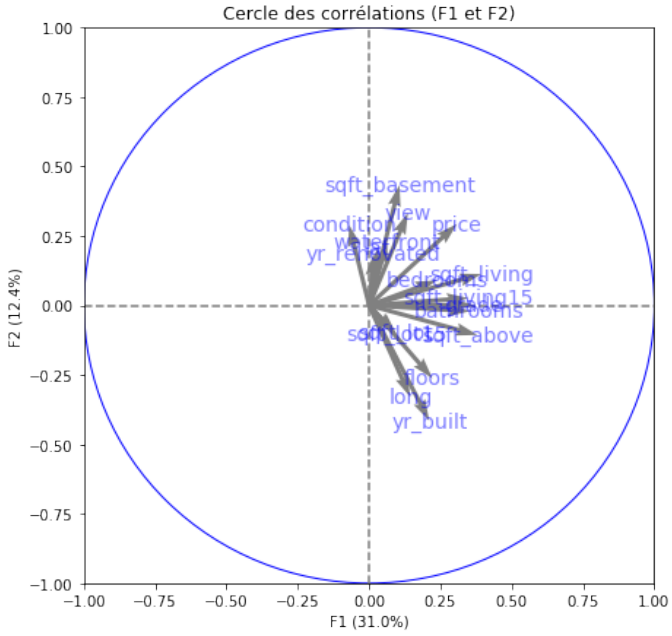


FIGURE 2.2 – Cercle des corrélations de F1 et F2

Interprétation : La plus part des variables sont très faiblement corrélées avec la première variable synthétique F1, mis à par waterfront, sqft_basement, sqft_lot, yr_renovated, view, condition et bathrooms qui sont tès peu représentées. Les variables les plus correlées avec F1 sont **sqft_above**, **bathrooms**, **sqft_living15** et **sqft_living**. Nous pouvons donc dire que la première variable synthétique représente les dimensios de la maison.

Les variables les plus correlées avec la variable synthétique F2 sont yr_built et sqft_basement. Nous povons donc dire que la deuxième variable synthétique représente la date de construction de la maison.

Variables synthétiques F3 et F4

Le cercle des corrélations des variables F3 et F4 est le suivant :

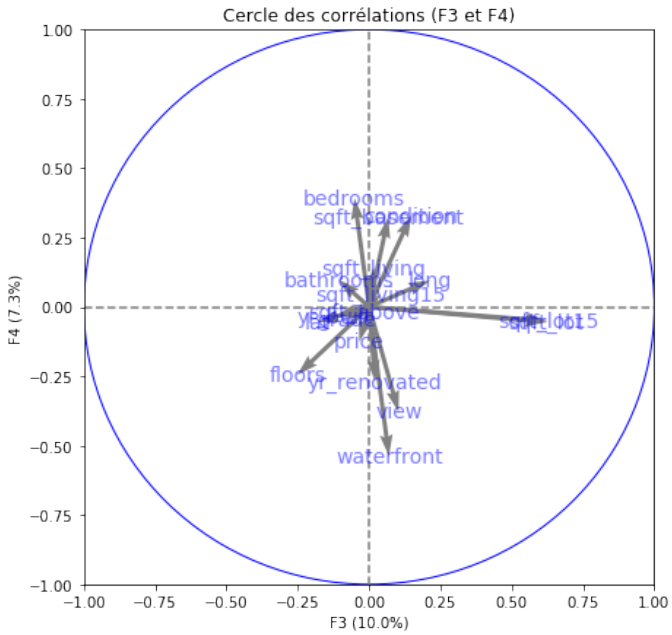


FIGURE 2.3 – Cercle des corrélations de F3 et F4

Interprétation : Les variables qui sont corrélées avec la troisième variable synthétique F3 sont sqft_lot et sqft_lot15. Nous pouvons donc dire que la variable synthétique F3 représente la superficie du terrain sur lequel se trouve la maison.
Cependant, les variables les plus correlées avec la variable synthétique F4 sont waterfront, view, et bedrooms. Nous pouvons dire que la Quatrième variable synthétique F4 représente le confort de la maison.

Variables synthétiques F5 et F6

Le cercle des corrélations des variables F5 et F6 est le suivant :

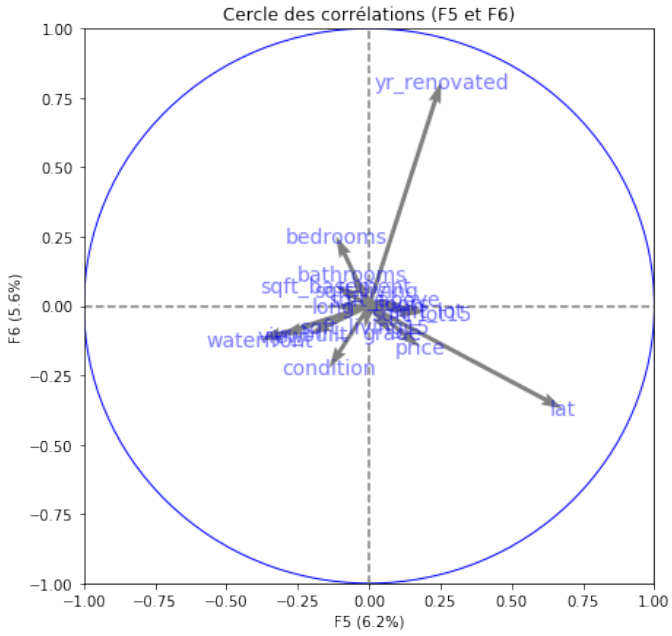


FIGURE 2.4 – Cercle des corrélations de F5 et F6

Interprétation : La seule variable qui est corrélée avec la Cinquième variable synthétique F5 est lat. Nous pouvons alors dire que la variable F5 représente la position de la maison dans la ville.
La variable la plus correlée avec la variable synthétique F6 est yr_renovated. La variable F6 représente alors l'année de rénovation des maisons.

2.1.2 Analyse de la variabilité entre les individus

L'analyse de la variabilité des individus consiste à étudier les ressemblances et les différences qui existent entre les différents individus. Pour ce travail, nous allons représenter les individus dans les plans factoriels. Comme défini dans la première partie, nous allons utiliser 6 composantes, donc nous aurons ici trois plans factoriels.

Projection des individus sur le premier plan factoriel

Nous allons ici projeter les individus sur le premier plan factoriel, c'est à dire sur un plan composé des deux premières composantes F1 et F2.

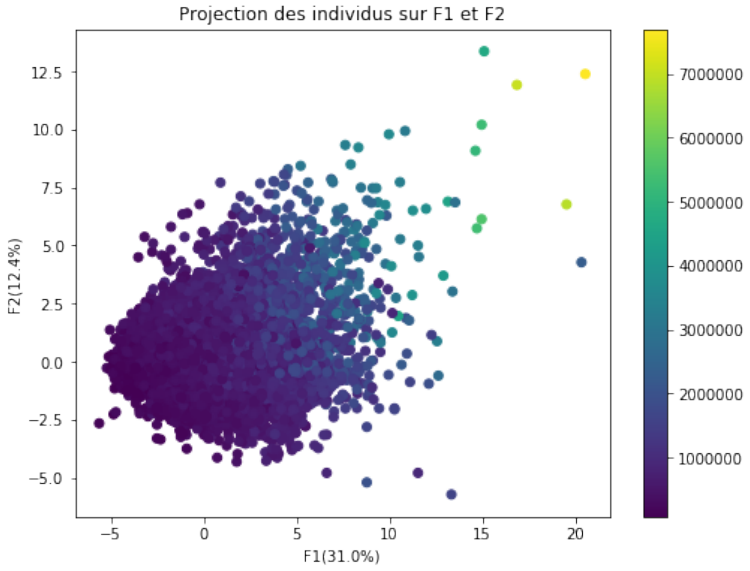


FIGURE 2.5 – Premier plan factoriel

Interprétation : En associant le premier cercle de corrélation, nous pouvons dire que la plus part des maisons de notre échantillon de population ont un une superficie moyenne. Nous remarquons que une minorité des maisons ont une grande superficie par rapport au autres.

Projection des individus sur le deuxième plan factoriel

Nous allons ici projeter les individus sur le deuxième plan factoriel, c'est à dire sur un plan composé des composantes F3 et F4.

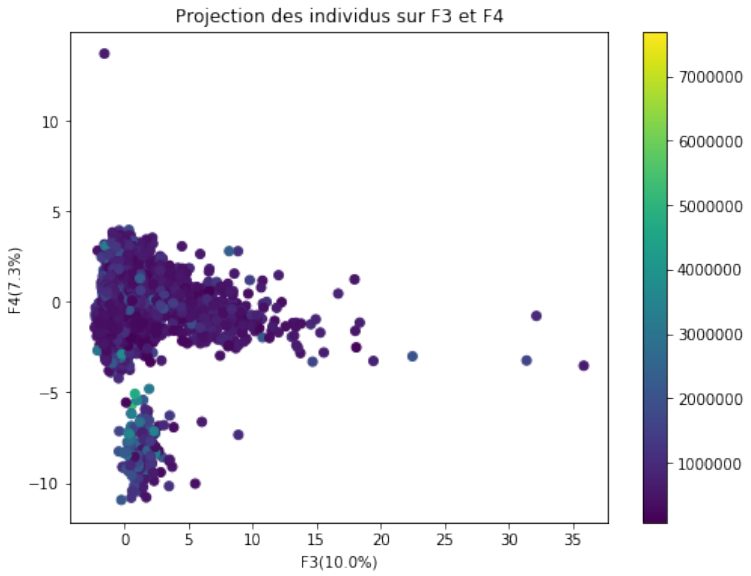


FIGURE 2.6 – Premier plan factoriel

Interprétation : En associant le deuxième cercle de corrélation, nous pouvons dire toutes les maisons ont en moyenne une superficie de terrain égales, mise à part certaines maisons qui ont une superficie de terrain plus grande que les autres. Nous pouvons également dire que la plupart des maisons on un confort assez équivalente.

Projection des individus sur le troisième plan factoriel

Nous allons ici projeter les individus sur le troisième plan factoriel, c'est à dire sur un plan composé des composantes F5 et F6.

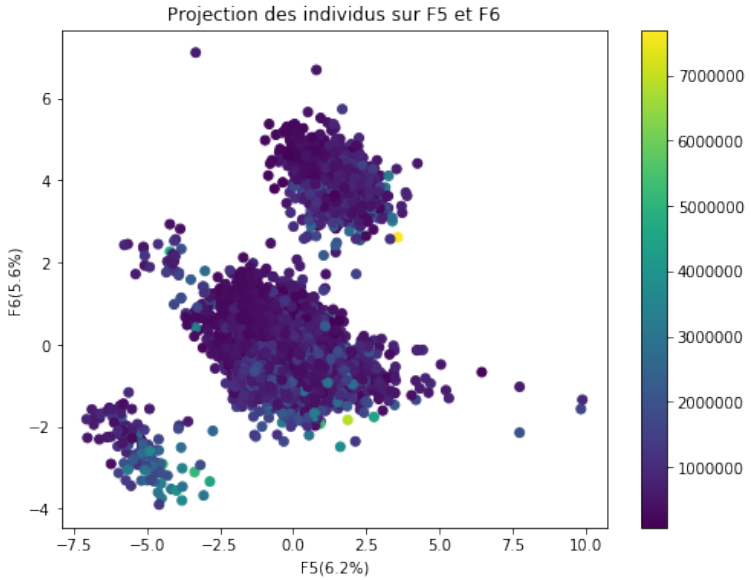


FIGURE 2.7 – Premier plan factoriel

Interprétation : En associant le troisième cercle de corrélation, nous pouvons dire que les maisons les plus à droite de notre shémas ont une bonne position geographique par rapport aux autres maisons. Nous remarquons aussi que plusieurs maisons ont été rénovées.

2.2 Classification (Clustering)

Avant d'éffectuer une classification sur notre jeu de données, nous allons réduire le nombre de nos variables de manière à au moins conserver 95% de leurs informations.

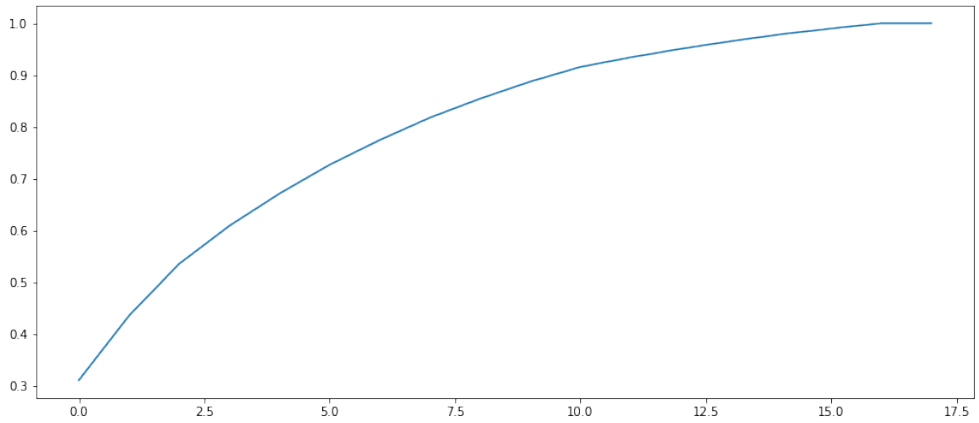


FIGURE 2.8 – Analyse de la représentativité des variables synthétiques

Après analyse de ce graphyque, nous pouvons conclure que nous pouvons réduire nos variables de 18 à 12 tout en conervant 95% de leurs informations.

Pour éffectuer la classification, nous allons utiliser la classification hiérarchique et l'algorithme de K-Means. Dans un premier temps, nous allons utiliser K-Means pour réduire nos données de 21 613 à 300 individus. Ensuite, nous allons appliquer la classification hiérarchique pour détecter le nombre de clusters et nous allons encore appliquer K-Means avec le nombre de cluter précédement trouvé.

Réduction du nombre d'individus

L'objectif ici est de réduire le nombre d'individus de notre jeu de données. Pour cela, nous allons utiliser l'algorithme K-Means avec 300 centroids. Ce travail nous permettra de passer de 21 613 individus à 300 individus. Nous affichons les 300 individus trouvés sur le premier plan factoriel.

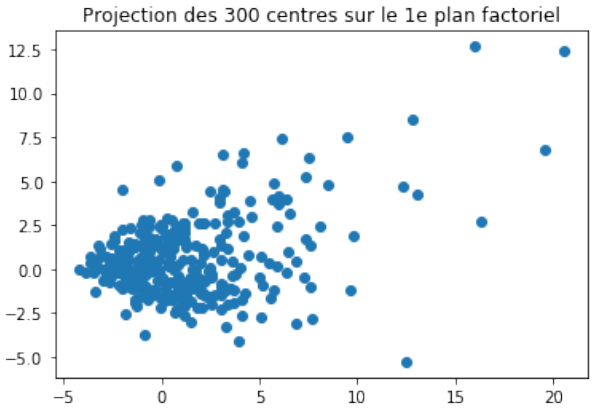


FIGURE 2.9 – Affichage des 300 centroids sur le premier plan factoriel

Classification hiérarchique

L'objectif de la classification hiérarchique est de regrouper les 300 individus obtenus précédement en classes. Après avoir appliqué l'algorithme de classification hiérarchique, nous obtenons le dendrogramme suivant.

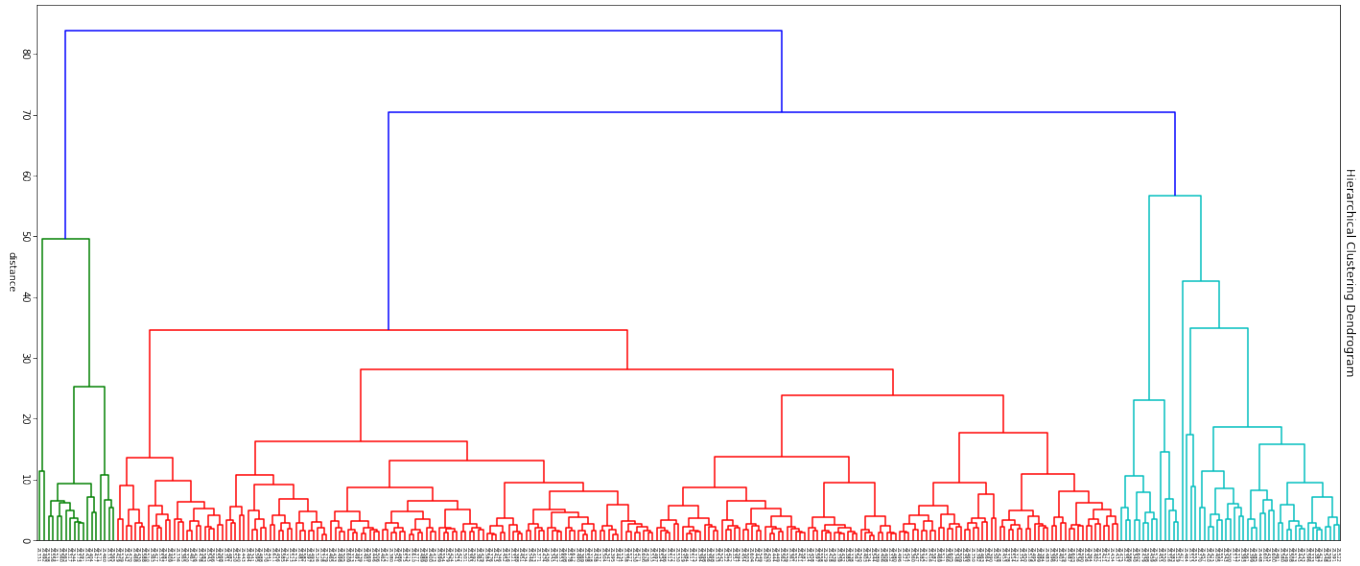


FIGURE 2.10 – dendrogramme - Classification hiérarchique

le dendrogramme obtenue nous montre que nous pouvons diviser nos individus en 3 classes.

Choix du nombre de classe pour la classification

Pour selectionner le nombre de classe à utiliser pour la classification, nous allons utiliser la méthode Elbow (méthode du coude).

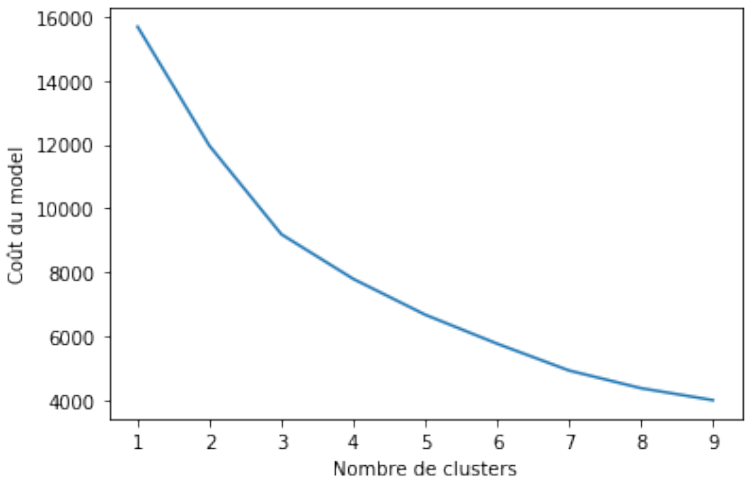


FIGURE 2.11 – dendrogramme - Classification hiérarchique

Après analyse de la diminution de la variance pour les valeurs de k comprisent entre 1 et 10, nous pouvons dire que le nombre optimal de clusters pour l'algorithme des K-Means est 3.

Classification des individus en 3 classes

Après la classification hiérarchique faite précédement, nous allons diviser nos individus en trois classes aec l'algorithmes K-Means. Après avoir divisé les individus, nous affichons les résultats obtenus sur les 3 Premiers plan factoriels.

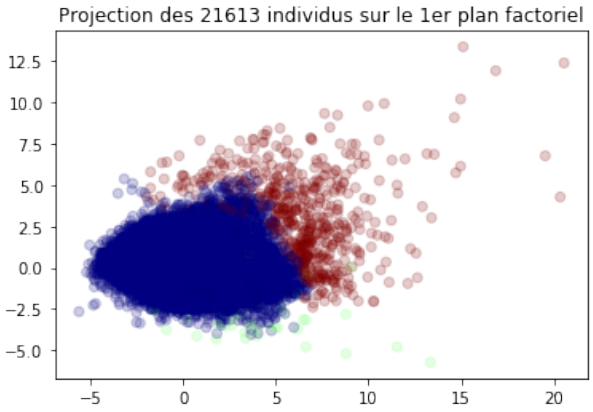


FIGURE 2.12 – Résultat K-means sur le 1er plan factoriel

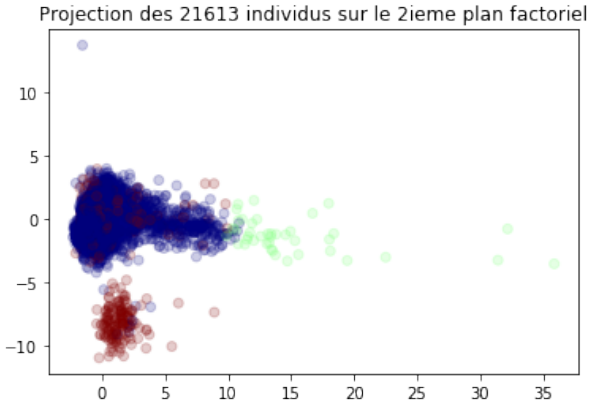


FIGURE 2.13 – Résultat K-means sur le 2e plan factoriel

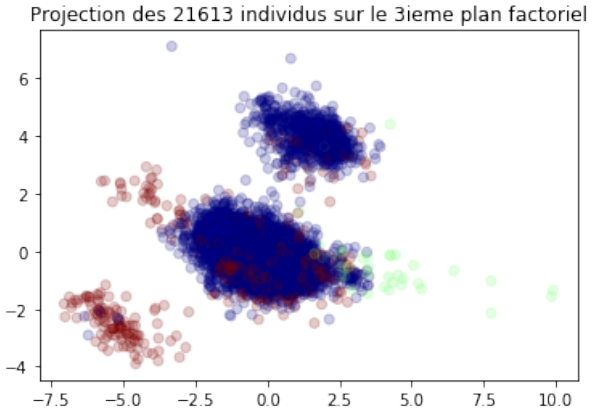


FIGURE 2.14 – Résultat K-means sur le 3e plan factoriel

Interprétation des résultats sur la classification

Interprétation par rapport aux plans factoriels

Nous pouvons dire que le premier groupe de maison (en vert) représente les maisons qui se trouvent sur une grande superficie de terrain. Ces maisons font également partie des maisons ayant la date de construction la plus ancienne et les mieux positionnée.

Nous pouvons dire que le deuxième groupe de maison (en rouge) représente les maisons qui sont proches de la mer et qui ont un standing moyen.

Nous pouvons dire que le troisième groupe de maison (en bleu) représente les maisons qui n'ont aucune particularité.

Interprétation sur la variable target

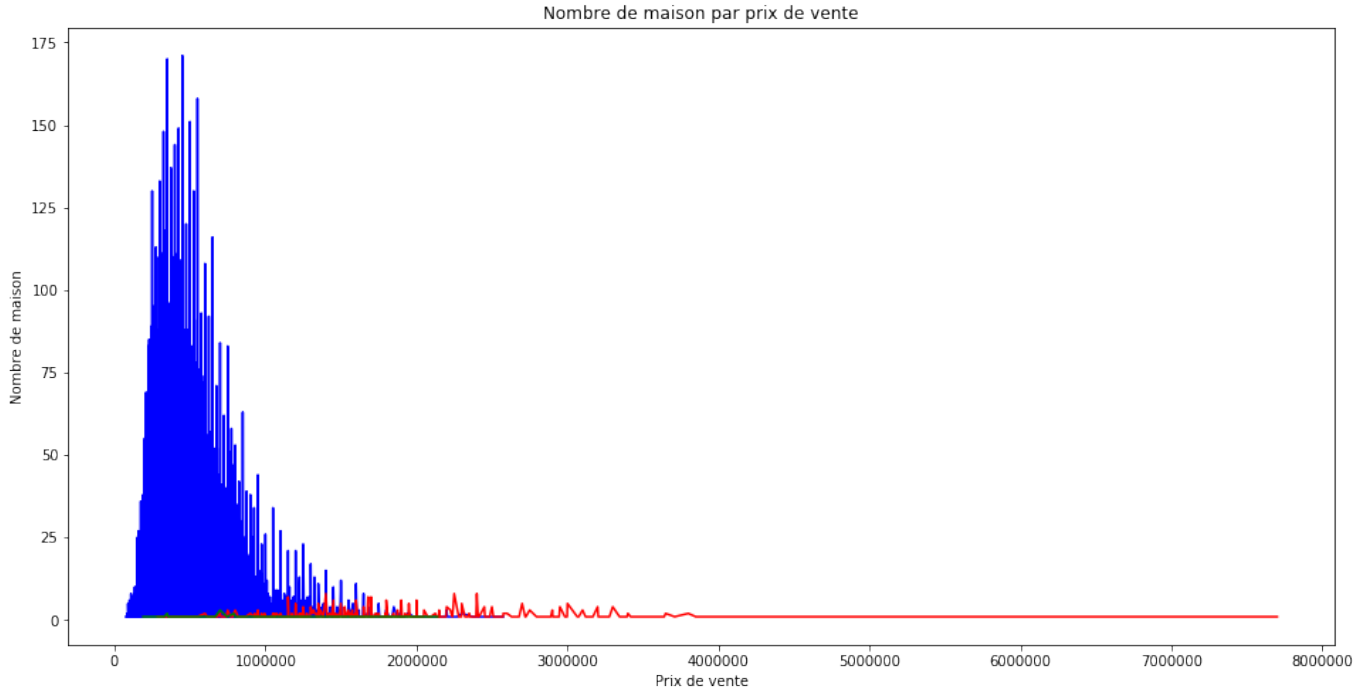


FIGURE 2.15 – Résultat K-means sur la variable target

Nous pouvons conclure que les maisons se trouvant proches de la mer sont les maisons les plus cher du marché.

CHAPITRE 3

APPRENTISSAGE SUPERVISÉ

Notre objectif ici est d'utiliser notre jeu de données pour pouvoir prédire le prix d'une maison en fonction de ses caractéristiques. Pour cela, nous allons faire de l'apprentissage supervisé et appliquer l'algorithme **SVM (Support Vector Machine)**. Nous allons utiliser la métrique de **Précision** pour évaluer le modèle que nous aurons créé en utilisant cet algorithme.

Nous allons dans un premier temps parler du fonctionnement de l'algorithme SVM et nous allons l'appliquer sur notre jeu de données. Par la suite, nous allons appliquer l'algorithme **KNN** et un **arbre de décision** afin de comparer les résultats obtenus avec SVM.

3.1 SVM

Les **SVM** (Supports Vectors Machines) ou séparateurs à vaste marge constituent un ensemble de techniques d'apprentissage surpervisé permettant de résourdre deux classes de problèmes : de **discrimination** et de **régression**.

Développés dans les années 1990 à partir des considérations théoriques **Vladimir Vapnik** et **Alexey Chervonenkis** qui tentent d'expliquer l'apprentissage d'un point de vue statistique.

Aujourd'hui très utilisés notamment dans les domaines de vision par ordinateur, de recherche d'information, de bio-informatique et des finances, les SVM doivent leur vulgarisation grâce à leur capacité à travailler avec de grandes dimensions de données, de faible nombre de hyperparamètre et de bons résultats parfois largement supérieur aux réseaux de neurones.

3.1.1 Formalisme des SVM

Les SVM sont utilisés pour résoudre des problèmes de dicrimination (**définr à quelle classe appartient un échantillon**) et de régression (**prédire la valeur numérique d'une variable**).

La resolution de tels problème consiste à déterminer un fonction objective définit par :

$$\mathbf{h} : \mathbf{y} = \mathbf{h(x)} \text{ où } \mathbf{x} \text{ est le vecteur d'entrée et } \mathbf{y} \text{ la sortie.}$$

Dans le cadre de notre travail, nous nous limiterons aux problèmes de discrimination binaire donc $\mathbf{y} \in \{\mathbf{1, -1}\}$ et $\mathbf{x}, \mathbf{w} \in R^N$.

On a définit la fonction **h** par : **h(x)**= w^T **x** + w_0. Alors **x** est de classe **1** si **h(x)** ≥ 0 et de classe **-1** sinon. La frontière de décision **h(x) = 0** est un hyperplan séparateur.

Il sera question de mener un apprentissage surpervisé sur notre fonction à partir d'un ensemble labelisé défini par les paire (x_k, l_j) sur un ensemble de taille **p**. Donc $l_k h(x_k) \geq 0$, avec 0 ≤ k ≤ p.

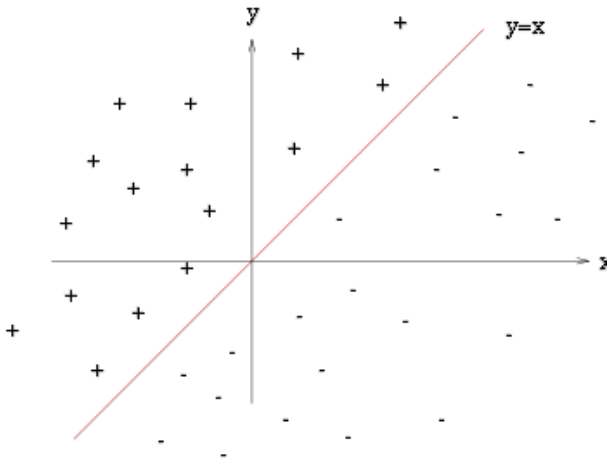


FIGURE 3.1 – Classifieur SVM des points + et -

3.1.2 Discrimination linéaire et hyperplan séparateur : problème lineairement séparable

Supposons un plan muni d'un ensemble de deux classes de points (+) et (-), nous aimerions determiner la frontière séparant le mieux ces classes. De telle sorte pour un nouveau point inconnu donné, défini uniquement par sa position ,nous puissons à l'aide de l'algorithme de classification prédire sa classe de points.

Proposition d'une solution possible :

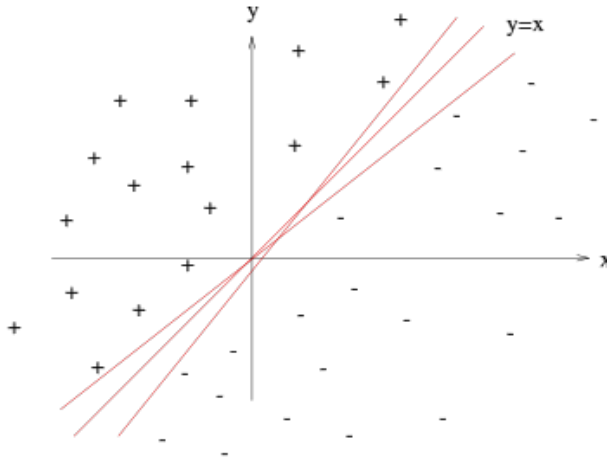


FIGURE 3.2 – Classifieur SVM des points + et - avec une infinité d'hyperplans

nous constatons pour ce problème de discrimination, il peut en exister une inifinité d'hyperplans séparant nos deux classes.

Question : Comment determiner l'hyperplan le plus optimal ?

Solution : Trouver un hyperplan qui maximise la marge.

La Marge maximale

La marge est la distance entre l'hyperplan et les échantillons les plus proches. Ces derniers sont appelés **vecteurs supports**.

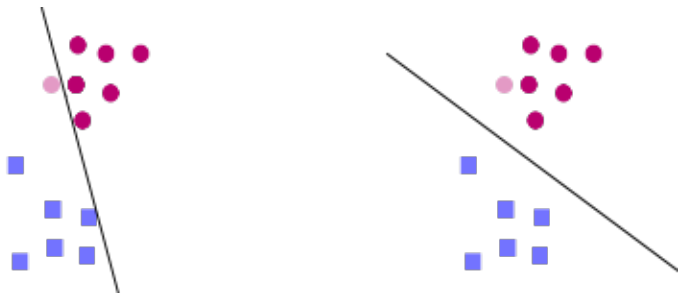


FIGURE 3.3 – capacité de généralisation est meilleure à droite (avec l'hyperplan optimal) qu'à gauche

L'hyperplan qui maximise la marge, c'est à dire une frontière aussi loin que possible de chacune des classes, est donné par :

$$\textbf{arg max min } \{||\mathbf{x} - x_k|| : \mathbf{x} \in R^N, w^T\mathbf{x} + w_0 = \mathbf{0}\}$$

Recherche de l'hyperplan optimal

Formulation primale

La marge est la plus petite distance entre les échantillons d'apprentissage et l'hyperplan séparateur satisfaisant la condition :

$$l_k(w^T\mathrm{x} + w_0) \geq 0$$

La distance d'un échantillon x_k est obtenue par sa projection orthogonale sur le vecteur de poids.

$$\frac{l_k(w^T x + w_0)}{||w||}$$

L'hperplan séparateur de marge maximale (w,w_0) est obtenu par :

$$\textbf{arg } \max_{w,w_0} \frac{1}{||w||} \min_k [l_k(w^T x_k + w_0)]$$

Pour optimiser, on choisit de normaliser les composant w et w_0 de telle sorte que :

$$\begin{cases} \mathrm{w}^T\ x_m^+ arge + w_0 = 1 \\ \mathrm{w}^T\ x_m^- arge + w_0 = \text{-}1 \end{cases} \quad donc \ \ \mathrm{l}_k(w^T\mathrm{x} + w_0) \geq 0$$

On obtient donc une forme canonique de l'hyperplan, donc la marge vaut désormais : $\frac{1}{||w||}$

La formulation dite primale des SVM s'exprime alors sous la forme suivante :

$$\text{Minimiser } \frac{1}{2}||w||^2 \quad \text{sous les contraintes } \quad l_k(w^T x_k + w_0) \geq 1$$

Une approche de résolution par la méthode classique des multiplicateurs de Lagrange, permet d'obtenir le lagrangien devant être minimisé par rapport à w et w_0, et maximisé par rapport à α.

$$L(w, w_0, \alpha) = \frac{1}{2}||w||^2 - \sum_{k=1}^{p} \alpha_k \left\{ l_k(w^T x_k + w_0) - 1 \right\} \qquad (1)$$

Formulation duale

Par annulation des dérivées partielles du lagrangien, selon les conditions de **Kuhn-Tucker**, on obtient :

$$\begin{cases} \sum_{k=1}^{p} \alpha_k l_k x_k & = w^* \\ \sum_{k=1}^{p} \alpha_k l_k & = 0 \end{cases}$$

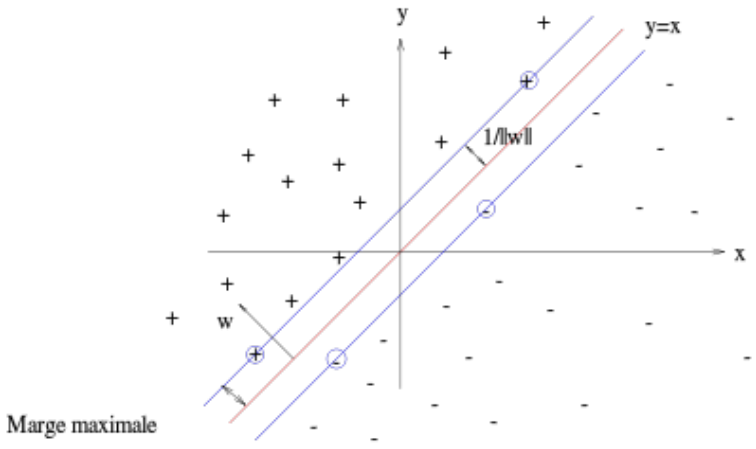
Et après remplacement dans (1) on a la formule duale suivante :

$$\text{Maximiser } \tilde{L}(\alpha) = \sum_{k=1}^{p} \alpha_k - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j l_i l_j x_i^T x_j \qquad (2)$$

On obtient des multiplicateurs de Lagrange optimaux. Afin d'obtenir l'hyperplan solution, on remplace w par sa valeur optimale w*, dans l'équation de l'hyperplan, ce qui donne :

$$h(x) = \sum_{k=1}^{p} \alpha_k^* l_k (x \cdot x_k) + w_0$$

Illustration :



L'hyperplan solution ne dépend que du produit scalaire entre le vecteur d'entrée et les vecteurs supports. Cela est à l'origine de innovation majeure des SVM : le passage par un espace de redescription grâce à une fonction noyau.

problème non séparable : kernel trick ou l'astuce du noyau

l'astuce du noyau est une méthode qui permet d'utiliser un classifieur linéaire pour résoudre un problème non linéaire. L'idée est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, où un classifieur linéaire peut être utilisé et obtenir de bonnes performances. La discrimination linéaire dans l'espace de grande dimension (appelé aussi espace de redescription) est équivalente à une discrimination non linéaire dans l'espace d'origine.

Plus formellement, on applique aux vecteurs d'entrée x une transformation non-linéaire.

L'espace d'arrivée de la fonction $\phi(\mathbf{x})$ est appelé espace de redescription.

$$\mathbf{h(x)} = w^T\phi(x) + w_0 \text{ avec } l_k h(x_k) \geq 0$$

En utilisant la même procédure que dans le cas sans transformation, on aboutit au problème d'optimisation suivant :

$$\text{Maximiser } \tilde{L}(\alpha) = \sum_{k=1}^{p} \alpha_k - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j l_i l_j \phi(x_i)^T\phi(x_j) \qquad (3)$$

$$\text{Sous les contraintes } \alpha_i \geq 0 \text{ , et } \sum_{k=1}^{p} \alpha_k l_k = 0$$

Le problème de cette formulation est qu'elle implique un produit scalaire entre vecteurs dans l'espace de redescription, de dimension élevée, ce qui est couteux en termes de calculs. Pour résoudre ce problème, on utilise une astuce connue sous le nom de Kernel trick, qui consiste à utiliser une fonction noyau, qui vérifie :

$$K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$$

d'où l'expression de l'hyperplan séparateur en fonction de la fonction noyau :

$$h(x) = \sum_{k=1}^{p} \alpha_k^* l_k K(x_k, x) + w_0$$

L'intérêt de la fonction noyau est double

— Le calcul se fait dans l'espace d'origine, ceci est beaucoup moins coûteux qu'un produit scalaire en grande dimension.
— La transformation n'a pas besoin d'être connue explicitement, seule la fonction noyau intervient dans les calculs. On peut donc envisager des transformations complexes, et même des espaces de redescription de dimension infinie.

choix du noyau

Le choix du noyau est dirigé par le **théorème de Mercer** qui montre qu'une fonction noyau K continue, symétrique et semi-définie positive peut s'exprimer comme un produit scalaire dans un espace de grande dimension.

L'approche par Kernel trick généralise ainsi l'approche linéaire. Le noyau le plus simple est le noyau linéaire :

$$K(x_i, x_j) = x_i^T \cdot x_j$$

Mais les noyaux usuels employés dans SVM sont :

— **le noyau polynômial** :

$$K(x_i, x_j) = (x_i^T \cdot x_j + 1)^d$$

— **le noyau gaussien**

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$$

— **le noyau laplacien**

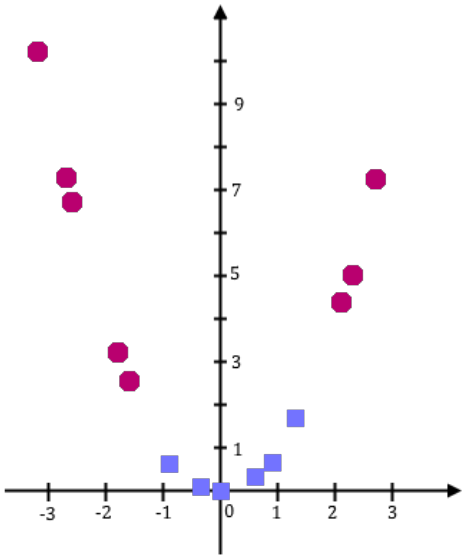
$$exp(-\frac{||x - x^{'}||}{\sigma})$$

— **le noyau rationnel**

$$1 - \frac{||x - x^{'}||^2}{||x - x^{'}||^2 + \sigma}$$

Illustration

Après redescription du problème dans un espace de dimension supérieur, on obtient en choisant ϕ(x) = ($\frac{x-150}{10}$, ($\frac{x-150}{10}$)^2) :



Discrimination muti-classe

supposons qu'on veuille entraîner un SVM à distinguer les ronds, les triangles et les losanges, nous avons besoin d'un SVM qui sache distinguer ces trois classes différentes.

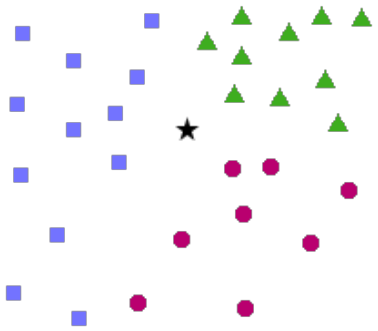


FIGURE 3.4 – prédire la classe du nouveau echantillon

Approche One vs one

Dans cette approche, on va créer des **Voteurs** : chaque voteur V_i, j détermine si mon entrée xx a plus de chances d'appartenir à la catégorie **i** ou à la catégorie **j**.

Ainsi un voteur V_i, j est un **SVM** un SVM qui s'entraîne sur les données de catégorie **i** et **j**.

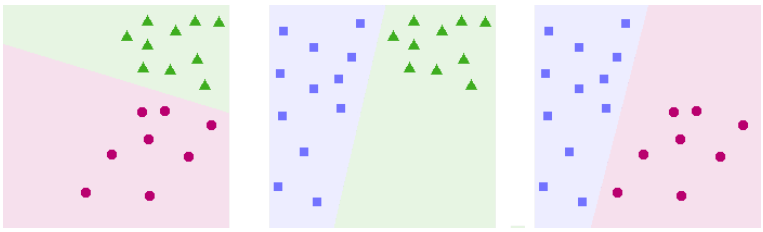


FIGURE 3.5 – Approche one by one

Pour classer une entrée on retournera tout simplement la catégorie qui aura remporté le plus de duels.

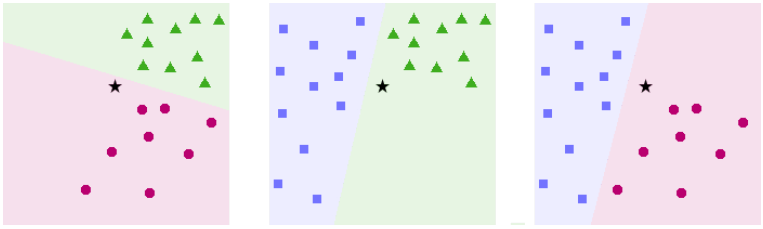


FIGURE 3.6 – prédire un échantillon par l'approche one by one

L'inconvénient de cette méthode est que le nombre de voteurs est proportionnel au carré du nombre de catégories : pour 10 catégories, ce sont 55 voteurs qu'il faut créer, mais pour 20 catégories, il en faudra 210, d'où un temps de calcul de plus en plus élevé.

Approche One vs all

L'approche one-vs-all consiste à créer un SVM par catégorie. Dans notre exemple, un SVM sera ainsi spécialisé dans la reconnaissance des ronds, un autre dans la reconnaissance des triangles, et un autre dans les losanges.

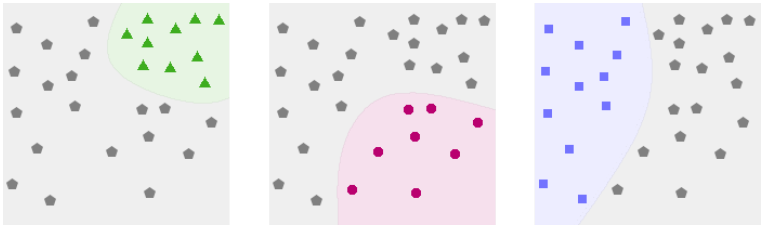


FIGURE 3.7 – Approche one by all

Pour classer une nouvelle entrée, on regarde à quelle catégorie la nouvelle entrée est le plus probable d'appartenir.

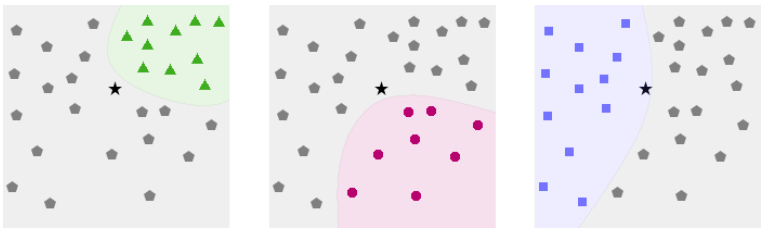


FIGURE 3.8 – prédire un échantillon avec l'approche one by all

Il arrive que plusieurs SVM aient un résultat positif, dans ce cas-là, on prend celui qui est le plus certain de son résultat. De même, quand tous les résultats sont négatifs, on prend alors la catégorie du SVM pour lequel l'entrée est le plus près possible de la frontière.

L'inconvénient de cette méthode est que, quand le déséquilibre entre la quantité de données dans les deux catégories est trop fort, un SVM obtient de moins bons résultats.

3.2 Utilisation de SVM sur notre jeu de données

Pour utiliser la machine à verteur de support, nous devons commencer par discrétiser notre variable price. Ensuite, nous allons chercher les meilleurs paramètres pour appliquer la machine à verteur de support, nous allons ensuite chercher la meilleure divion possible de notre jeux de données et nous allons finir par tracer la courbe d'apprentissage de notre modèle.

3.2.1 Discrétisation de la variable price

Pour éffectuer la Discrétisation, nous allons utiliser la formule de Sturges pour avoir une approximation du nombre de classe que nous devons avoir.

$$\text{nbre\_classe} = \log2(21613 + 1) = 14{,}4$$

Nous aurons donc environs 15 classes après la discrétisation de la variable price. Maintenant, nous calculons l'étendue des différentes classes afin de déterminer les intervalles. Pour cela, nous divisons l'intervalle des prix (max - min) par le nombre de classe.

$$\text{largeur} = (\text{max} - \text{min}) \,/\, \text{nbre\_classe} = (7700000 - 75000) \,/\, 15 = 508333.33$$

Nous pouvons donc dire que nos différentes classes aurons une étendue de **500 000**. Nous pouvons à présent créer nos différentes classes :

Classe	**Etiquette**	**Intervalle**
1	1	[0, 500000 [
2	2	[500000, 1000000 [
3	3	[1000000, 1500000 [
4	4	[1500000, 2000000 [
5	5	[2000000, 2500000 [
6	6	[2500000, 3000000 [
7	7	[3000000, 3500000 [
8	8	[3500000, 4000000 [
9	9	[4000000, 4500000 [
10	10	[4500000, 5000000 [
11	11	[5000000, 5500000 [
12	12	[5500000, 6000000 [
13	13	[6000000, 6500000 [
14	14	[6500000, 7000000 [
15	15	[7000000, 7500000 [
16	16	[7500000, ... [

TABLE 3.1: Liste des classes de la variable price

Nous obtenons ainsi la répartition de la variable price suivante :

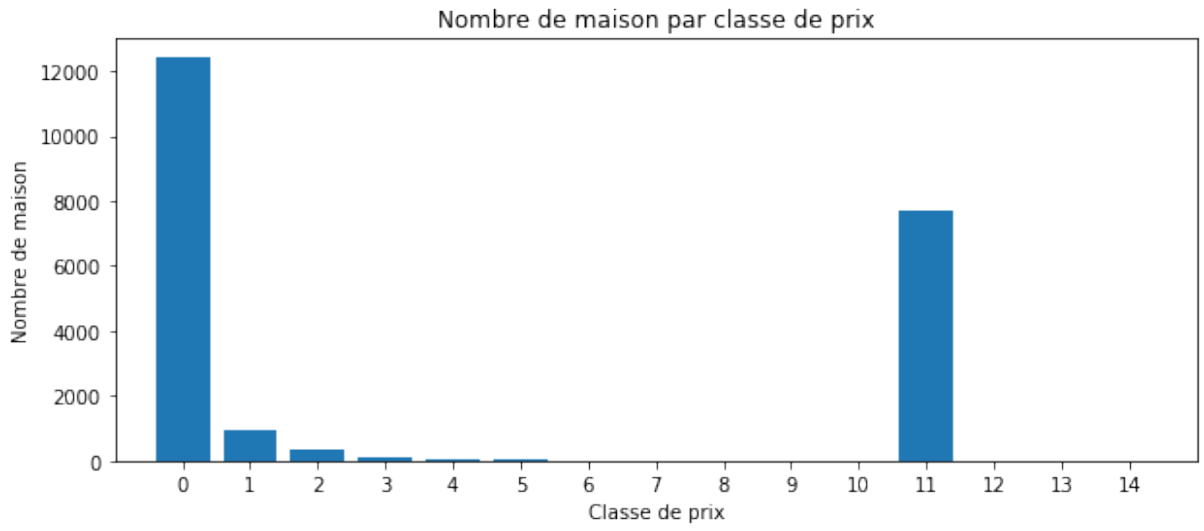


FIGURE 3.9 – Variable price discrétisée

3.2.2 Utilisation de SVM

Pour effectuer l'apprentissage supervisé sur notre jeu de données, nous allons commencer par définir un pipeline, ensuite, nous allons chercher les hyper-paramètres qui vont nous donner les meilleurs résultats. Nous allons ensuite tester le modèle que nous aurons créé avec plusieurs divisions possibles du jeu de données en partie d'entrainement et partie de test. Nous allons enfin interpréter les résultats obtenues.

Définition du pipeline

Notre pipeline comprend 2 principales parties :

— **le pré-processing** : Cette phase consiste à faire un pré-traitement sur les données. Elle se divise elle même en deux parties :
  — **Le centrage et la réduction des données** : Elle s'applique aux variables ayant des valeurs continues. Elle permet de mettre la moyenne de chaque variable à 0 et son écart-type à 1.
  — **l'encodage OneHot** : Elle s'applique aux variables ayant des valeurs discrètes.
— **l'estimateur** : Cette phase consiste à appliquer l'algorithme SVM une fois le pré-processing terminé.

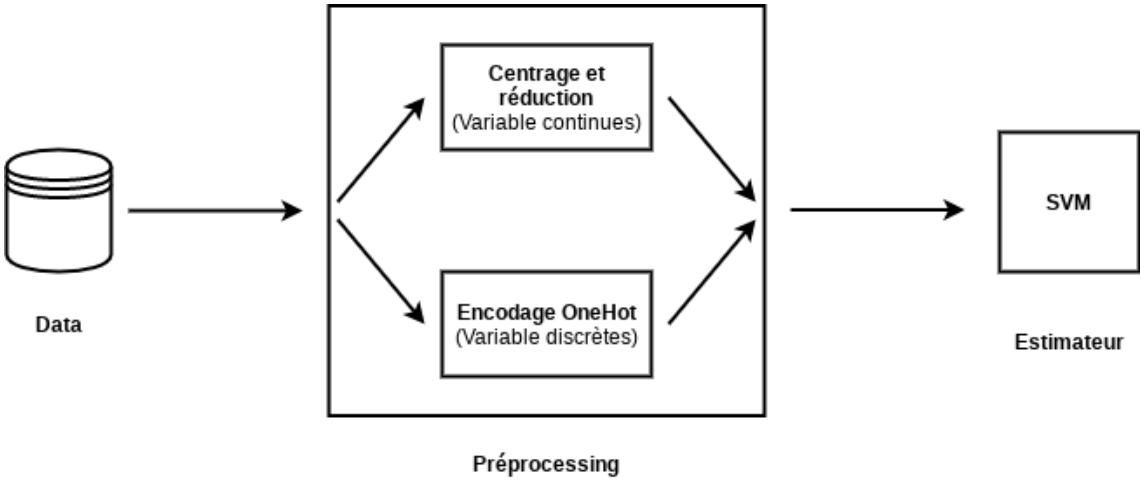


FIGURE 3.10 – Pipeline SVM

Recherche des meilleurs paramètres pour le modèle

Les principaux paramètres à régler pour SVM sont

— **kernel** : Noyau à utiliser (Les valeurs possibles sont **rbf**, **poly**, **linéaire**, **sigmoïde**)
— **gamma** : Coefficient du noyau (Les valeurs possibles sont **scale**, **auto**)
— **decision_function_shape** : Indique s'il faut retourner une fonction de décision (Les valeurs possibles sont **ovo** et **ovr**)

Après le test des différentes valeurs de paramètres, nous obtenons les meilleurs résultats pour les valeurs suivantes :

— **kernel** : **rbf**
— **gamma** : **auto**
— **decision_function_shape** : **ovo**

Recherche du meilleur découpage des données

Notre objectif ici est de trouver le meilleur découpage des données pour obtenir la meilleure Précision de notre modèle. Après un test des différents découpages possibles entre 10% et 60% de proportion pour les données de test, nous obtenons le tableau suivant :

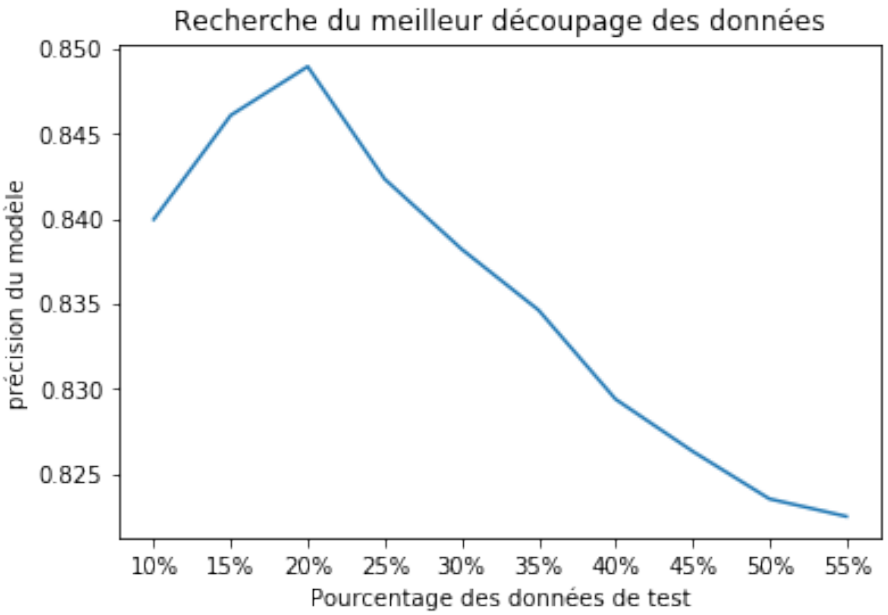


FIGURE 3.11 – répartition des données

On se rend compte que la meilleure répartition des données est de 20% pour les données de test et 80% pour les données d'entrainement. Avec cette répartition, on obtient une précision de **84,89%**

Courbe d'apprentissage

Notre éffectif ici est d'évaluer la courbe d'apprentissage pour voir si l'on pourrait avoir de meilleurs résultat de précision de notre modèle avec plus de données. Après division de notre jeux de données en lot de 10 et en ajoutant chaque lot au fur et à mesure, nous obtenons la courbe d'apprentissage suivante :

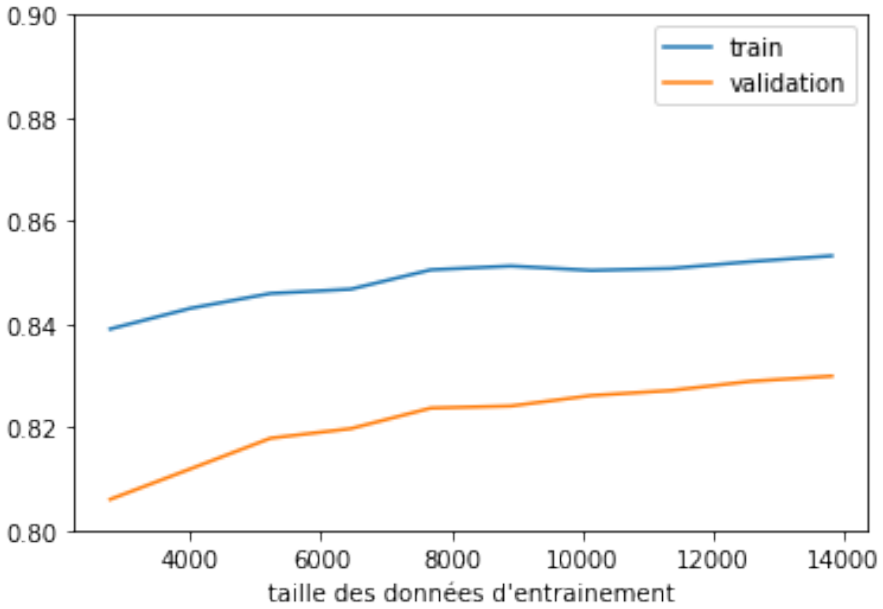


FIGURE 3.12 – Courbe d'apprentissage

Après évaluation de cette courbe, nous constatons que la précision augmente toujours lorsque le nombre

d'enrégistrement augmente. nous concluons que avec plus de données, nous pourrons avoir de meilleurs résultats pour la précision de notre modèle.

3.3 Utilisation d'autres méthodes d'apprentissage supervisé

Nous allons utiliser 2 autres méthodes pour comparer les résultats avec SVM que nous avons utilisé précédement. Les méthodes que nous allons utiliser sont **KNN** et **Arbre de décision**

3.3.1 KNN (k-nearest neighbors)

Généralités sur les KNN

KNN est un algorithme d'apprentissage supervisé qui classe un nouveau point de données dans la classe cible, en fonction des caractéristiques de ses points de données voisins

Ils peuvent être utilisé pour la classification et la regression.

— en classification k-NN, le résultat est une classe d'appartenance. Un objet d'entrée est classifié selon le résultat majoritaire des statistiques de classes d'appartenance de ses k plus proches voisins, (k est un nombre entier positif généralement petit). Si k = 1, alors l'objet est affecté à la classe d'appartenance de son proche voisin.

— en régression k-NN, le résultat est la valeur pour cet objet. Cette valeur est la moyenne des valeurs des k plus proches voisins.

Définition du pipeline

Le pipeline que nous allons utiliser est pareil que celui utilisé précédement, à la seule différence de l'estimateur qui est maintenant KNN.

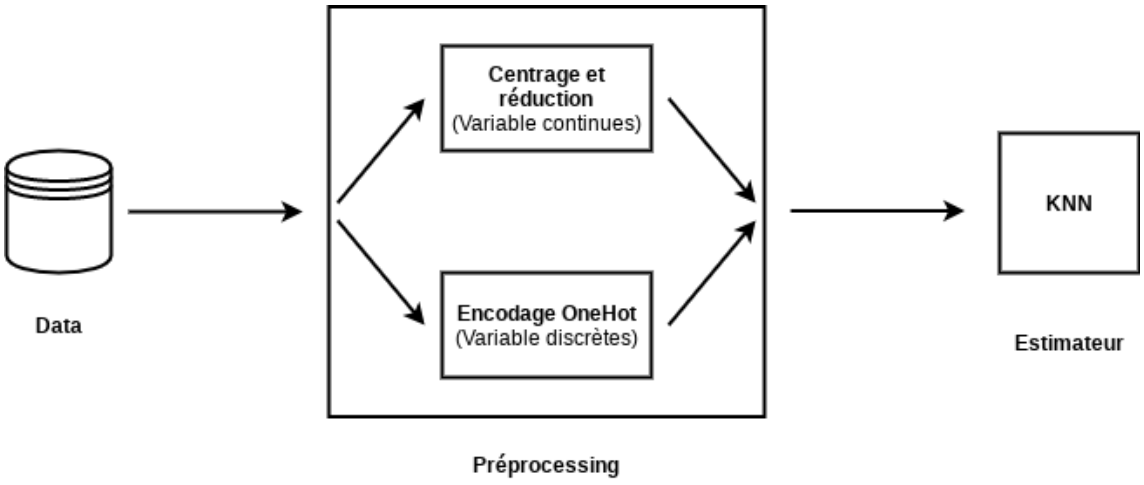


FIGURE 3.13 – Pipeline KNN

Recherche des meilleurs paramètres pour le modèle

Le paramètre que nous allons évaluer ici est le nombre de voisins à utiliser dans l'algorithme. Pour cela, nous testons les différentes valeurs possibles avec le nombre de voisins allant de 1 à 20. Nous obtenons le graphique suivant.

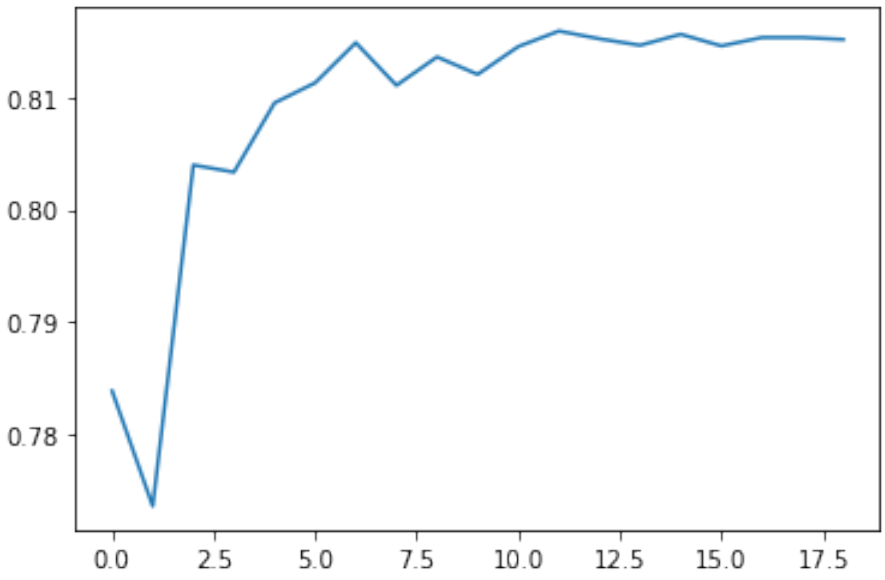


FIGURE 3.14 – Recherche de paramètre optimal

Nous pouvons donc conclure que le nombre de voisins à utiliser est 7.

Recherche du meilleur découpage des données

Notre objectif ici est de trouver le meilleur découpage des données pour obtenir la meilleure Précision du modèle KNN. Après un test des différents découpages possibles entre 10% et 60% de proportion pour les données de test, nous obtenons le tableau suivant :

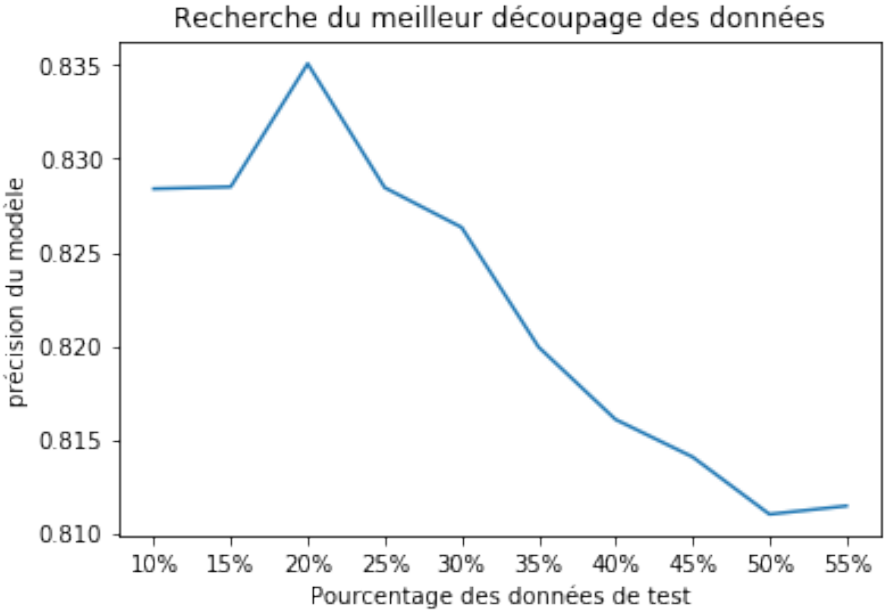


FIGURE 3.15 – répartition des données pour KNN

On se rend compte que la meilleure répartition des données est de 20% pour les données de test et 80% pour les données d'entrainement. Avec cette répartition, on obtient une précision de **83,51%**

Courbe d'apprentissage

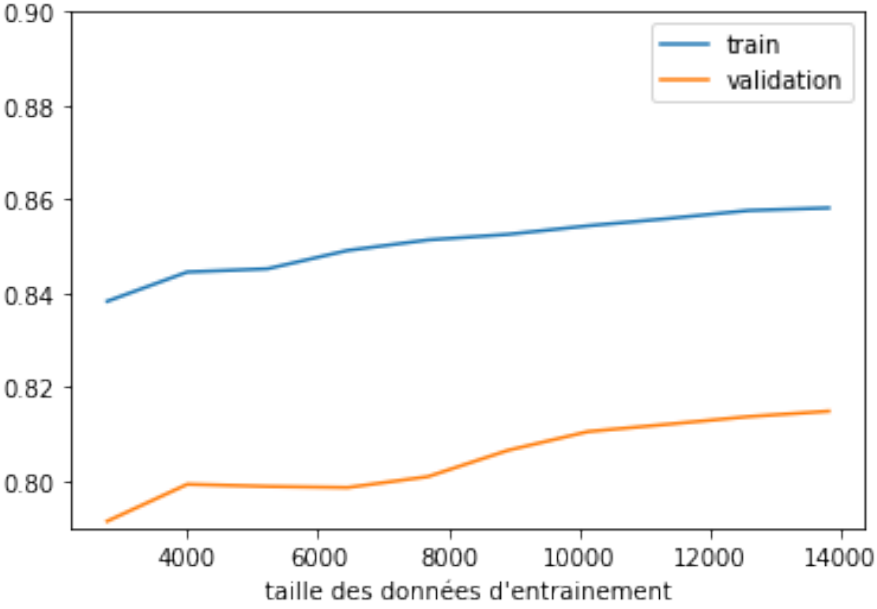


FIGURE 3.16 – Courbe d'apprentissage KNN

Après évaluation de cette courbe, nous constatons que la précision augmente toujours lorsque le nombre d'enrégistrement augmente. nous concluons que avec plus de données, nous pourrons avoir de meilleurs résultats pour la précision du modèle KNN.

3.3.2 Arbre de décision

Généralités sur les arbres de décision

Il s'agit d'un algorithme d'apprentissage automatique supervisé qui ressemble à un arbre inversé, dans lequel chaque nœud représente une variable prédictive (caractéristiques), le lien entre les nœuds représente une décision et chaque nœud feuille représente un résultat. Les arbres de décision font partie des algorithmes les plus populaires en apprentisssage automatique.

Les arbres de décision existent sur deux variantes :

— Les arbres de classification (Classification Tree) permettent de prédire à quelle classe la variable-cible appartient, dans ce cas la prédiction est une étiquette de classe

— Les arbres de régression (Regression Tree) permettent de prédire une quantité réelle (par exemple, le prix d'une maison ou la durée de séjour d'un patient dans un hôpital), dans ce cas la prédiction est une valeur numérique

Cependant les arbres utilisés dans le cas de la régression et dans le cas de la classification présentent des similarités mais aussi des différences, en particulier en ce qui concerne la procédure utilisée pour déterminer les séparations des branches.

Définition du pipeline

Le pipeline que nous allons utiliser est pareil que celui utilisé précédement, à la seule différence de l'estimateur qui est maintenant DecisionTreeClassifier.

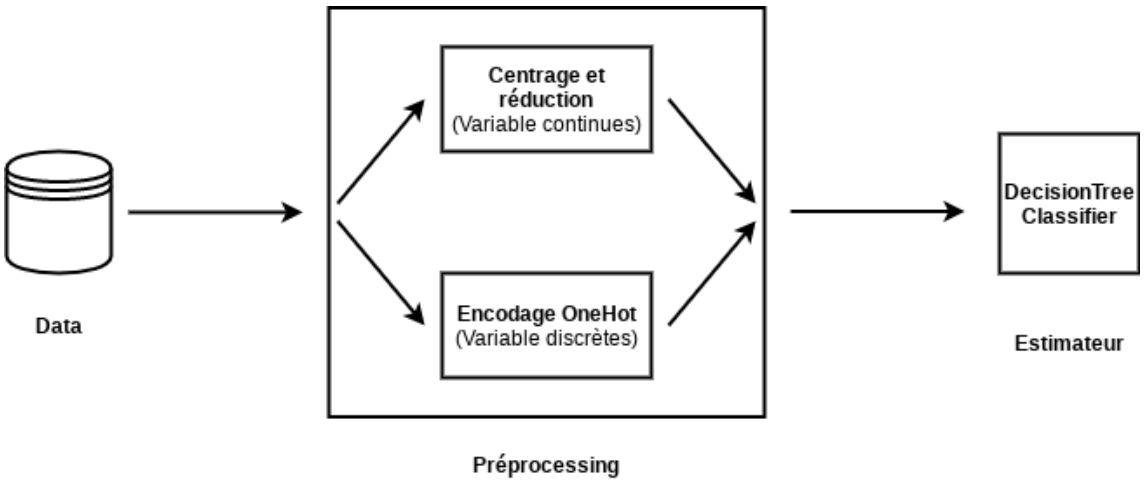


FIGURE 3.17 – Pipeline arbre de décision

Recherche des meilleurs paramètres pour le modèle

Le paramètre que nous allons évaluer ici est le nombre de voisins à utiliser dans l'algorithme. Pour cela, nous testons les différentes valeurs possibles pour le paramètre criterion. ce paramètre peut prendre les valeurs **gini** et **entropy**

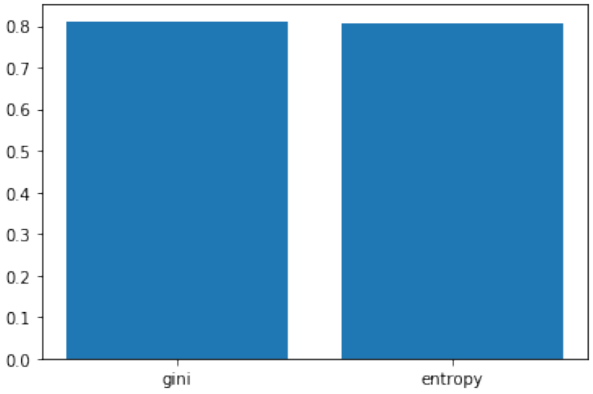


FIGURE 3.18 – Recherche de paramètre optimal

Nous constatons que les deux valeurs nous donne le même résultat. Nous allons alors choisir la valeur **gini**.

Recherche du meilleur découpage des données

Notre objectif ici est de trouver le meilleur découpage des données pour obtenir la meilleure Précision du modèle KNN. Après un test des différents découpages possibles entre 10% et 60% de proportion pour les données de test, nous obtenons le tableau suivant :

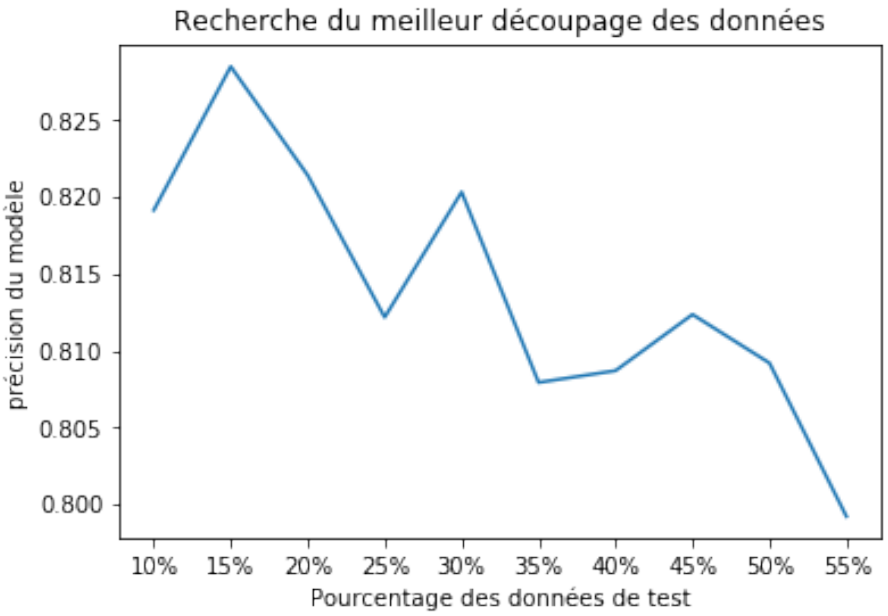


FIGURE 3.19 – répartition des données pour KNN

On se rend compte que la meilleure répartition des données est de 15% pour les données de test et 85% pour les données d'entrainement. Avec cette répartition, on obtient une précision de **82,45%**

3.3.3 Régression linéaire

Généralités sur la régression linéaire

Le but de la régression linéaire est d'expliquer une variable continue Y à l'aide d'une variable X. La variable Y est appelée variable dépendante, ou variable à expliquer et la variable X est appelée variable indépendante, ou variable explicative.

Nous allons appliquer ici une **régression linéaire multiple** parce que notre variable X est constituée de plusieurs sous-variables explicatives. Etant donné que la variable dépendante doit être continue, la discrétisation effectué précédement n'est pas appliquer dans ce contexte.

Définition du pipeline

Le pipeline que nous allons utiliser légèrement diférent de ceux utilisé dans les méthodes précédente.

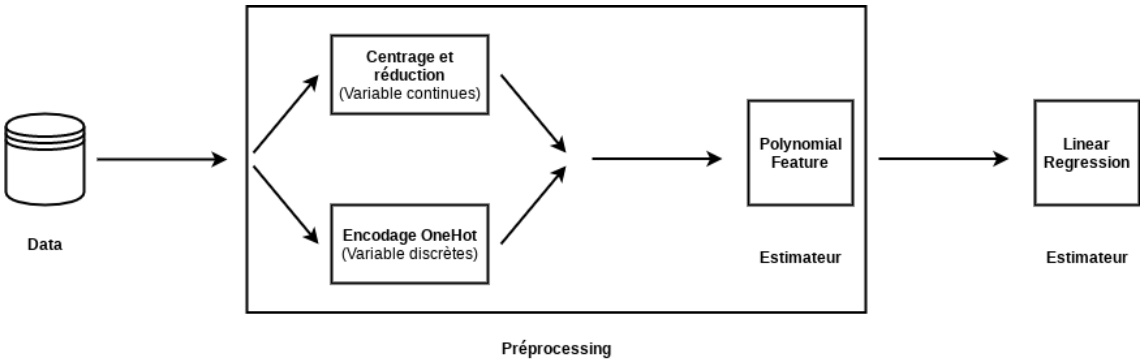


FIGURE 3.20 – Pipeline régression linéaire

Recherche des meilleurs paramètres pour le modèle

Le paramètre que nous allons évaluer ici est le dégré de la fonction polynomiale. Nous essayons pour cela les valeurs 1, 2 et 3.

Après avoir tester les différentes valeurs, nous constatons que la meilleur valeur de dégré polynomiale est la valeur **1**.

Recherche du meilleur découpage des données

Notre objectif ici est de trouver le meilleur découpage des données pour obtenir la meilleure Précision du modèle de régression. Après un test des différents découpages possibles entre 10% et 60% de proportion pour les données de test, nous obtenons le tableau suivant :

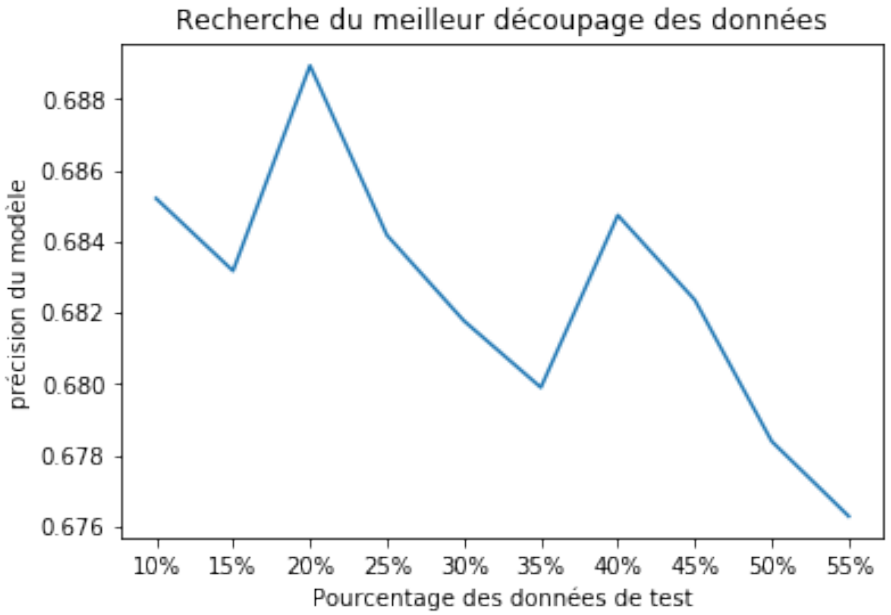


FIGURE 3.21 – répartition des données pour la régression linéaire

On se rend compte que la meilleure répartition des données est de 20% pour les données de test et 80% pour les données d'entrainement. Avec cette répartition, on obtient une précision de **68,89%**

3.4 Comparaison des différents modèles utilisés

Pour l'apprentissage suprvisé, nous avons utilisé 3 modèles : **SVM**, **KNN** et **Arbre de décision**. Le tableau suivant permet de comparer les résultats obtenus avec ces différents modèles.

Modèle	Meilleurs paramètres	Précision
SVM	kernel : rbf gamma : auto decision_function_shape Test : 20%	**84,89%**
KNN	nombre de voisin : 7 Test : 20%	**83,51%**
Arbre de décision	criterion : gini Test : 15%	**82,45%**
Régression linéaire	degre : 1 Test : 20%	**68,89%**

TABLE 3.2: Comparaison des différents modèles utilisés

Nous pouvons donc conclure que le modèle qui nous donne de meilleures perfomances est le modèle **SVM** avec une précision de **84,89%**. Mais comme nous l'avons vue plus haut, nous pouvons améliorer ces perfomances si nous avons plus de données.

Cependant, le modèle avec la régression linéaire est préférable à tous les autres modèle à cause de la discrétisation de la variable price qui n'est pas raisonable du point de vue des utilisateurs. Pour éviter la disparité et se rapprocher le plus possible des valeurs de prix réelles, il est préférable d'utiliser le modèle avec la régression linéaire avec un score de **68,89%**.

CONCLUSION

Parvenu au terme de l'étude des données dont nous disposons ou il était question pour nous de faire une analyse descriptive et exploratoire, appliquer les méthodes d'apprentissage non supervisé et les méthodes d'apprentissage supervisé, nous pouvons dire que les objectifs fixés dès le bédut ont été atteint. Les données à notre disposition nous ont permis d'explorer les différentes facettes de la Fouille de données et découvrir les différents algorithmes qui existent. Notre analyse sur l'apprentissage supervisé nous a fait voir que nous pouvons obtenir de meilleurs résultats si nous disposons d'une quantité plus importante de données. Notre objectif après le travail fait dans ce rapport est allors de chercher à completer les données à disposition.