UNIVERSITÉ DE NANTES

IAE NANTES
ÉCONOMIE & MANAGEMENT

Modèles SVM & ANN sous python

Thibaud MEYNIER & Yasmina AMDJHADI

Dossier d'évaluation pour le Master 2

Econométrie et statistiques appliquées, pour le cours de Mr. Abrahamson

Master 2 EKAP

IAE Nantes

14/12/2020

Table des matières

1 Introduction

Le Machine Learning est une application de l'intelligence artificielle qui propose aux systèmes informatiques la possibilité d'apprendre et de s'améliorer automatiquement à partir d'une expérience effectuée au préalable. Cette méthode se concentre sur le développement de programmes informatiques qui peuvent accéder aux données et les utiliser pour apprendre d'eux même, afin d'être appliqués sur des données inédites. L'objectif principal est de permettre aux algorithmes d'apprendre automatiquement sans intervention humaine ni assistance, et d'ajuster les actions en conséquence. Il existe différentes méthodes d'apprentissages automatiques. Ces dernières sont souvent classées comme supervisées ou non supervisées. Dans notre étude nous allons nous intéresser plus précisément aux algorithmes d'apprentissages supervisés, afin de répondre à notre problématique. Notre étude répond à une compétition kaggle dont l'intitulé est : "House Price, advanced regression". Elle consiste à prévoir les prix de différents biens immobiliers. Pour ce faire, nous disposons de diversses informations, tel le nombre de salles de bains, l'années de construction ou encore la qualité des pièces. Dans le cadre de se dossier nous utiliserons deux méthodes de classification : les machines à support vectoriels (SVM) et les réseaux de neurones (ANN). Nous allons chercher à estimer du mieux possible ces prix à l'aide d'un modèle robuste de machine learning. Afin de faire de la classification, nous avons classifier la variable des prix immobiliers en 3 classes distinctes présentées ensuite.

Notre étude se présente de la manière suivante : tout d'abord nous allons expliquer brièvement la partie théorique des différents algorithmes supervisés que nous allons utiliser afin de répondre à la problématique. Dans une seconde partie, nous effectuons des statistiques descriptives de nos données. Dans une troisième partie, nous appliquerons ces méthodes à notre échantillon. Enfin nous discuterons des réultats et du meilleur modèles obtenus.

2 Méthodologies utilisées

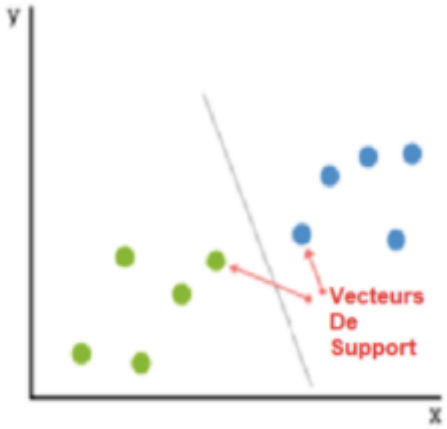
Les algorithmes d'apprentissages supervisés peuvent appliquer ce qui a été appris dans le passé à de nouvelles données à l'aide d'exemples étiquetés pour prédire les événements futurs. C'est à dire qu' à partir de l'analyse d'un ensemble de données d'apprentissage connu, l'algorithme d'apprentissage produit une fonction déduite afin de faire des prédictions sur les valeurs de sortie. Ainsi le système est capable de fournir des objectifs pour toute nouvelle entrée après une formation satisfaisante. En théorie, il est aussi capable de comparer sa sortie avec la sortie correcte et de trouver des erreurs afin de modifier le modèle en conséquence. Nous allons dans cette partie présenter quelques techniques d'apprentissage supervisés.

2.1 Les SVM

Les SVM linéaires

La méthode des SVM est un algorithme qui peut être utilisé à des fins de classification et de régression. Généralement elles sont utilisées dans les situations de classification. De plus, c'est une méthode qui peut être adaptée à une relation linéaire ou non linéaire. De manière simple, les SVM reposent sur l'idée de trouver un hyperplan qui divise au mieux un jeu de données en deux. Voici ci-dessous une représentation graphique de cette méthode pour une relation linéaire.

FIGURE 1 – Exemple de séparateur linéaire

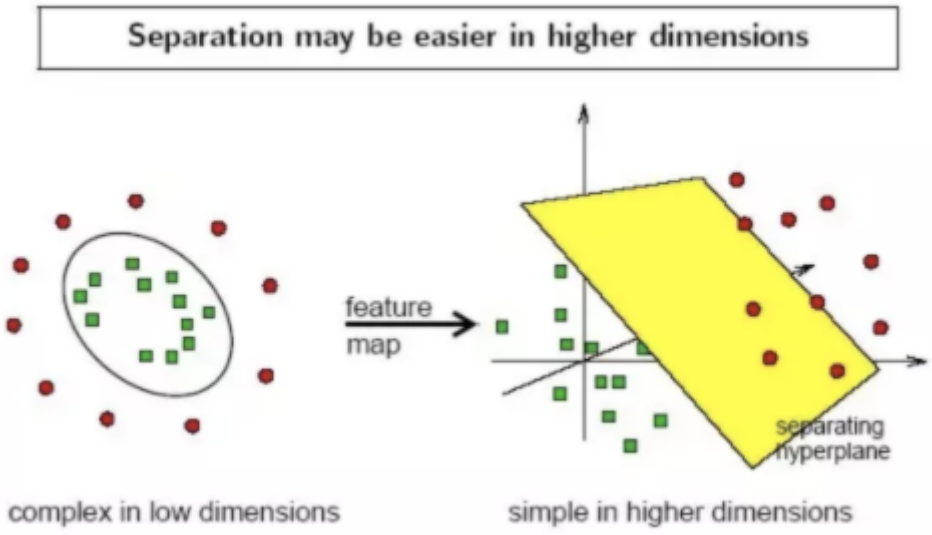


Les SVM non linéaires

Dans le cas des SVM non linéaires, le problème est un peu plus complexe. En effet, les données ne sont pas toujours aussi propres et faciles à traiter. Il peut arriver que les données soient tellement mélangées qu'il en devient impossible de le séparer de manière linéaire. Ainsi, pour classer un jeu de données de ce type, il est nécessaire de passer à plan tridimensionnel. Cette agrandissement de l'espace de fonctionnalités est connu sous le nom de Kernel (noyau).

La séparation des classes est donc plus facile et nous pouvons constater ceci dans le graphique suivant :

FIGURE 2 – Exemple de séparateur non linéaire



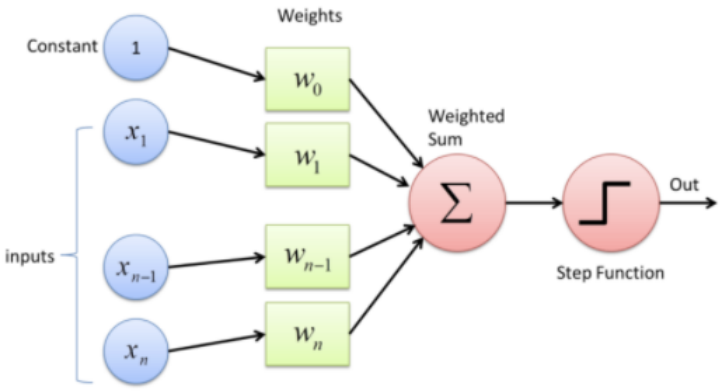
2.2 Les ANN

Cet algorithme est l'un des tout premiers algorithmes de Machine Learning. C'est un algorithme d'apprentissage supervisé de classificateurs binaires, séparant deux classes. Il s'agit d'un type de classifieur linéaire et du type de réseau de neurones artificiels le plus simple.

Dans le cas d'un classifieur binaire il ne peut y avoir que deux catégories de classification, et dans le cas d'un classifieur linéaire, les données d'entraînement doivent être classifiées en catégories correspondantes de sorte que toutes les données d'entraînement doivent être ordonnées dans ces deux catégories.

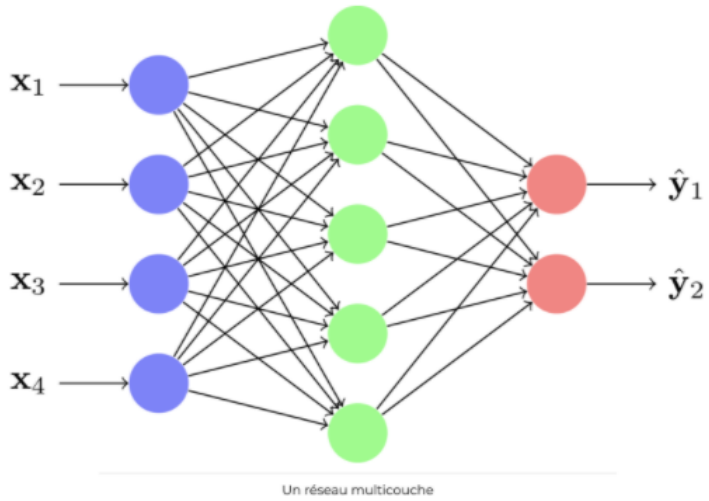
Bien que cet algorithme ait développé l'intelligence artificielle de façon majeure, les limites techniques ont rapidement été atteintes. En effet, un perceptron à une seule couche peut uniquement séparer les classes si elles sont séparables de façon linéaire.

FIGURE 3 – Exemple de réseaux de neurone simple



De ce fait, il a été découvert, par la suite, qu'un perceptron multicouche pouvait permettre de classifier des groupes qui ne sont pas séparables de façon linéaire. Il permet donc de résoudre des problèmes que les algorithmes à une seule couche ne peuvent pas résoudre. C'est un type de réseau formel qui s'organise en plusieurs couches. L'information circule de la couche d'entrée vers la couche de sortie. Tandis qu'un modèle monocouche ne dispose que d'une seule sortie pour toutes les entrées. Ainsi, chaque couche se compose d'un nombre de neurones variables, et les neurones de la dernière couche sont les sorties globales. C'est donc un réseau dit feedforward car il possède différents types d'interconnexions, une couche d'entrée et plusieurs couches actives. Il est donc capable d'approximer toute fonction continue, à condition que l'on fixe convenablement le nombre de neurones dans la couche cachée.

FIGURE 4 – Exemple de réseaux de neurones multicouches



3 Statistiques descriptives

Dans le tableau ci-dessous sont représentées les statistiques descriptives de la variables à expliquer SalePrice. Ainsi nous transformons cette variable quantitative en une variable catégorielle prenant 3 modalités :

— 0 si P≤163 000

— 1 si 163 000<P≤214 000

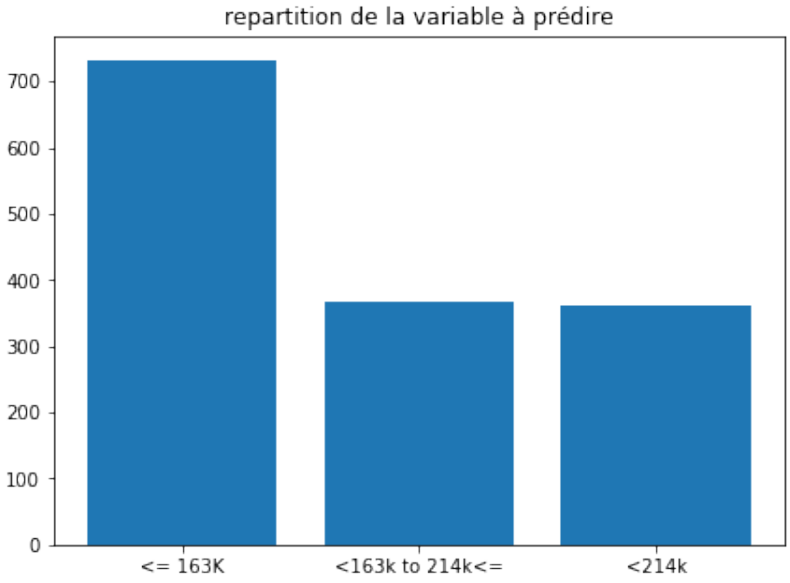
— 2 si P< 214 000

TABLE 1 – Statistiques descriptives

Count	1460
Mean	180 921,195
Std	79 442, 502
Min	34 900
1er quantile	129 975
Médiane	163 000
3eme quantile	214 000
Max	755 000

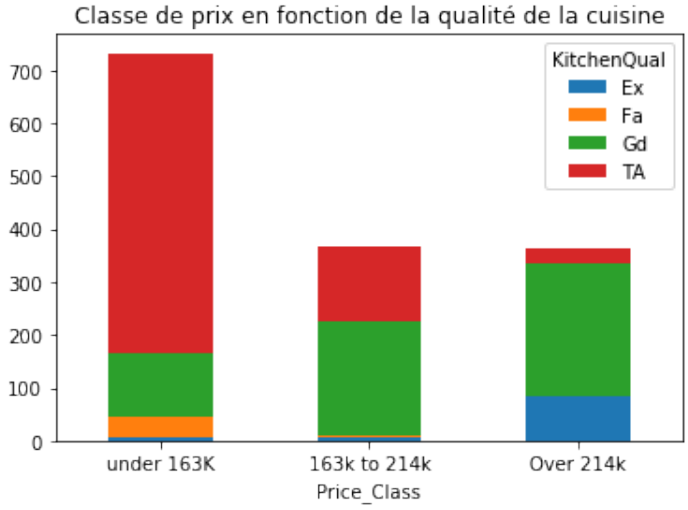
Dans le graphique ci-dessous, on observe la répartition de la variable explicative en 3 classes :

FIGURE 5 – Répartition de la variable à prédire



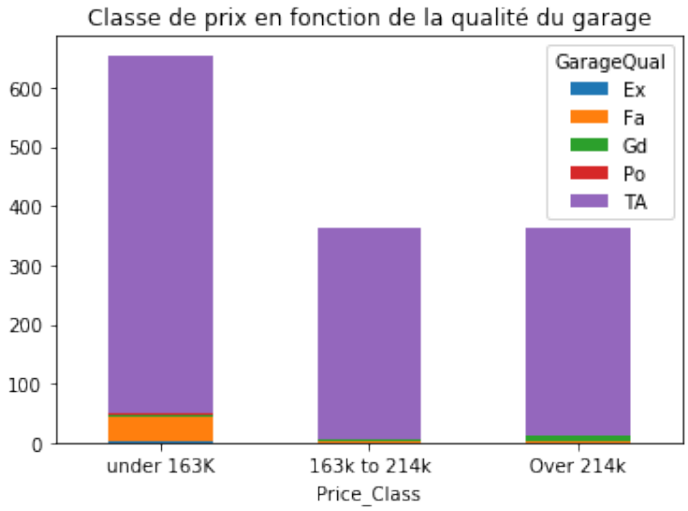
Nous présentons maintenant les graphiques croisés de notre variable à expliquer, Price_class et nos variables explicatives.

FIGURE 6 – Classe de prix en fonction de la qualité de la cuisine



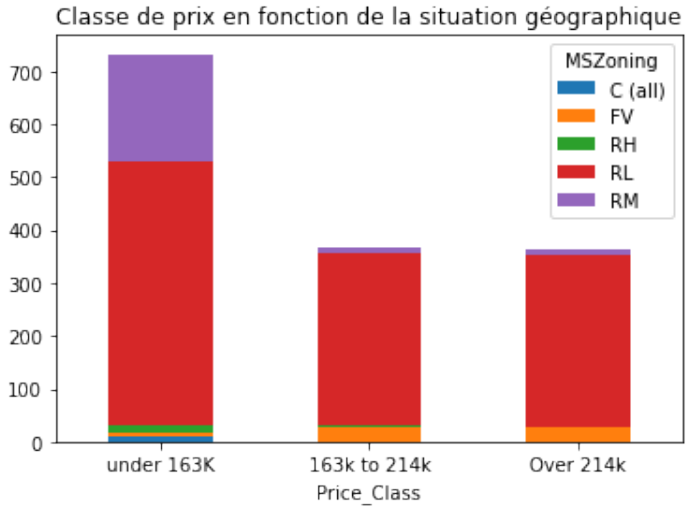
On constate qu'il existe une relation entre la qualité de la cuisine et la classe du prix du bien. Plus la qualité est bonne, plus la classe de prix est grande.

FIGURE 7 – Classe de prix en fonction de la qualité du garage



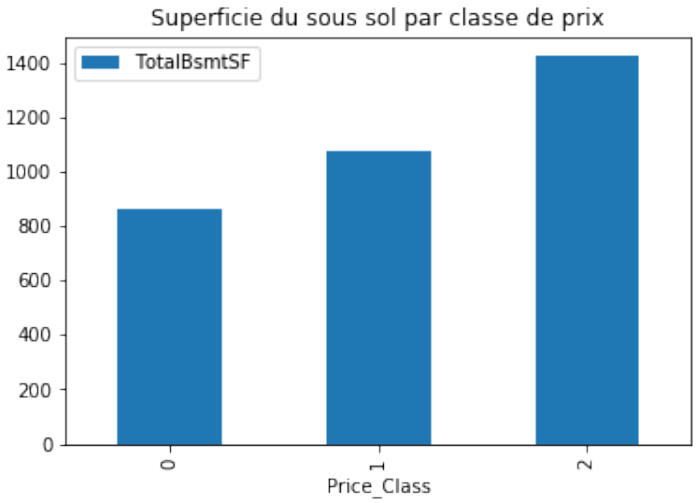
La qualité du garage ne semble pas impacter la classe de prix, on n'observe pas un fort pouvoir prédictif.

FIGURE 8 – Classe de prix en fonction de la situation géographique



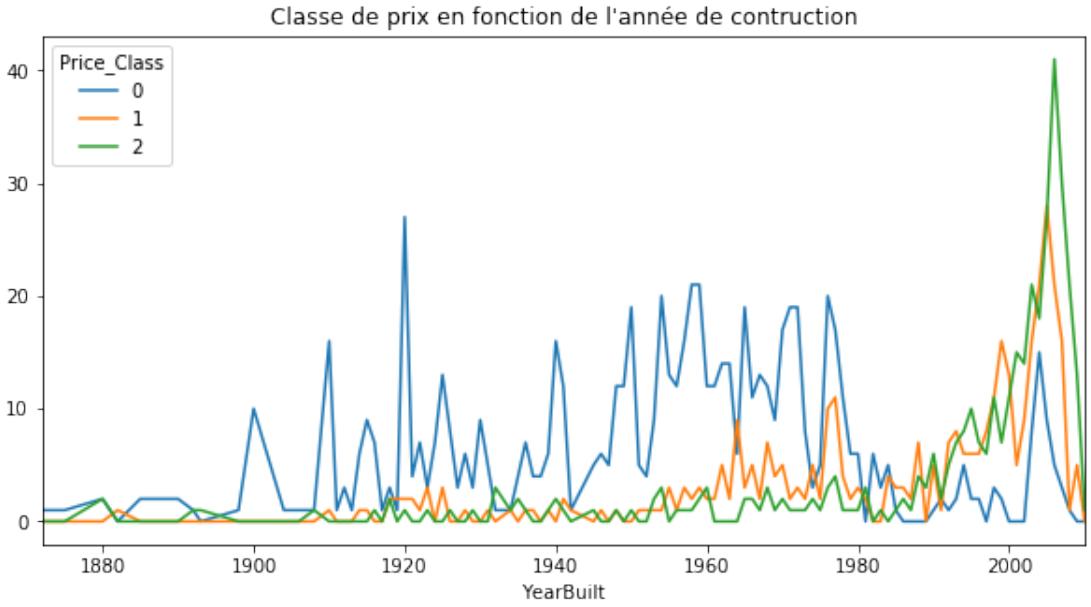
La situation géographique du bien semble impacter légèrement la classe de prix.

FIGURE 9 – Classe de prix en fonction de la superficie du sous sol



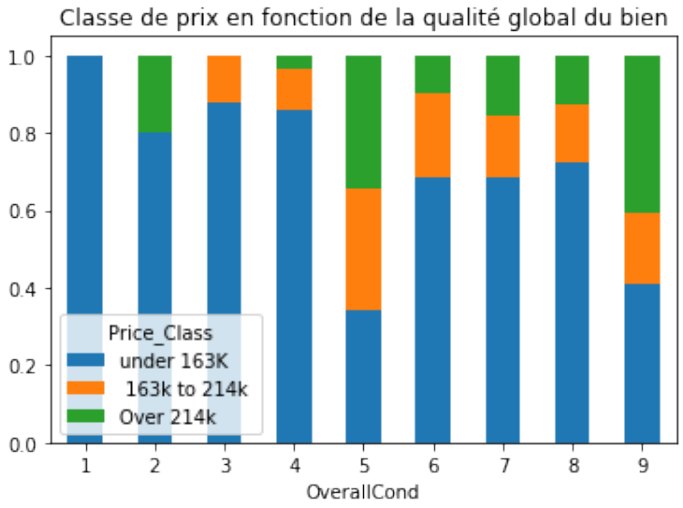
Il y a une relation positive logique et un impact peut être non linéaire de cette variable sur le prix. Le kernel gaussien en SVM pourra peut être permettre de gagner en qualité de prédiction par rapport au kernel linéaire.

FIGURE 10 – Classe de prix en fonction de l'année de construction



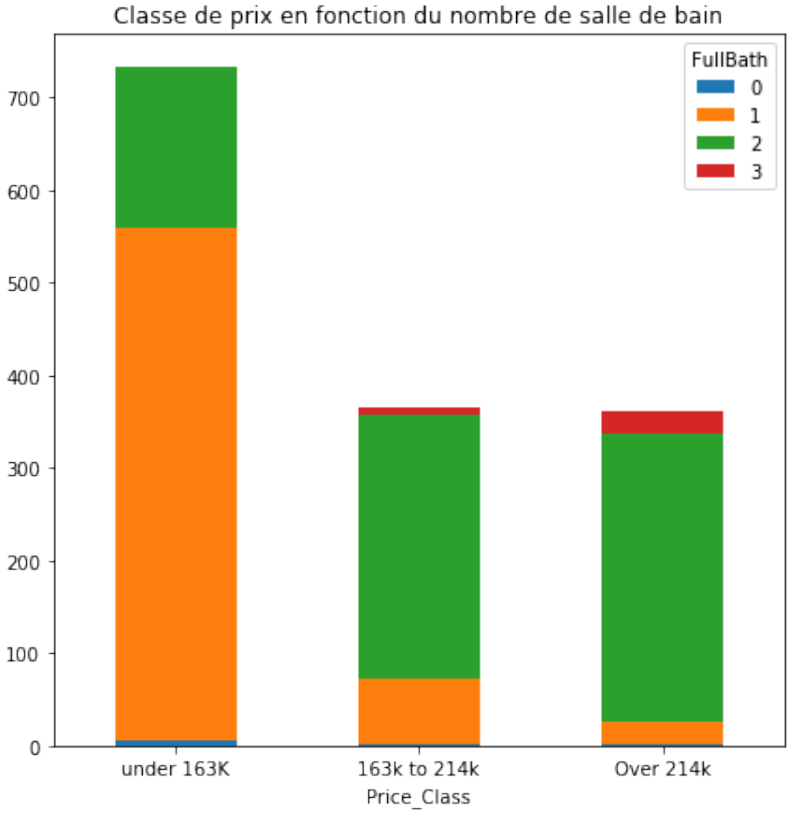
On constate qu'il y a beaucoup de biens de classe 1 construits avant les années 1980.

FIGURE 11 – Classe de prix en fonction de la qualité globale du bien



On observe que quand la qualité globale du bien est inférieur à 5, la catégorie de prix sera en dessous ou égale à 163 000 en majorité. Les classes 1 et 2 sont plus présente en proportions dans les notes de qualité globale élevé.

FIGURE 12 – Classe de prix en fonction du nombre de salle de bain



Il y a une relation croissante entre le nombre de salle de bains et la classe de prix.

4 Modélisations

Nous souhaitons maintenant sélectionner l'algorithme qui va prédire le mieux la réponse à notre problématique. Pour cela nous avons nettoyé et ciblé les données d'entraînement.

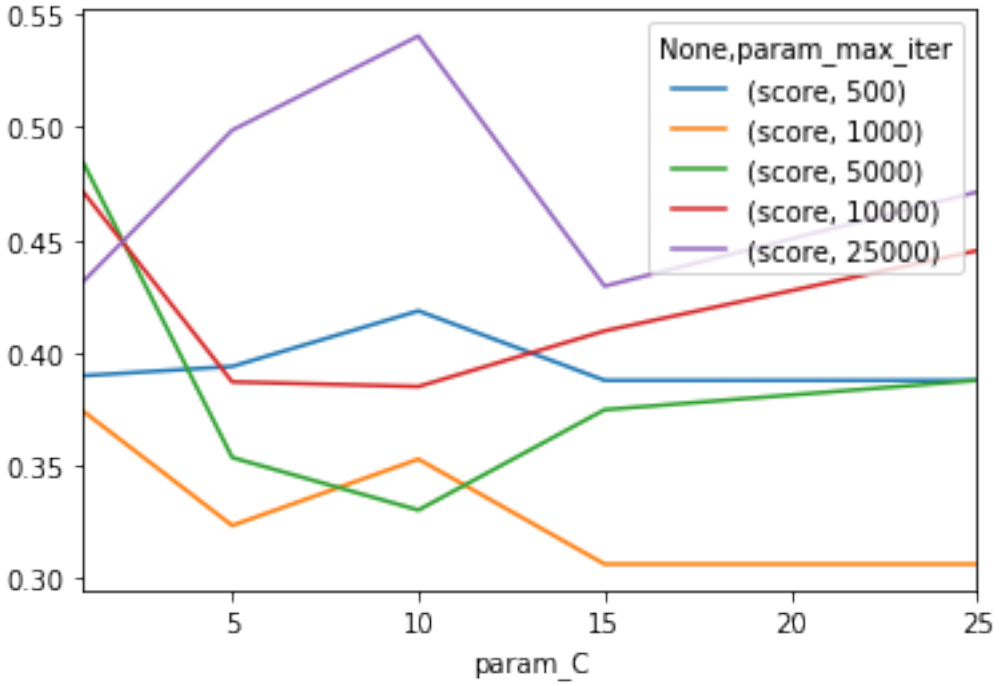
4.1 Modèles SVM

4.1.1 *SVM linéaire*

Dans cette partie nous souhaitons appliquer la méthode des SVM linéaires sur notre jeu de données. Pour cela nous utilisons la fonction SVC de la bibliothèque "sklearn". Dans notre cas, le SVM vise à séparer les observations selon leur classe de prix de vente. Sur le même principe que celui décrit dans la partie 1, nous testons les performances du modèle sur les données de l'échantillon "in sample" et "out of sample". Pour cela, nous spécifions les hyper-paramètres (C et Gamma), c'est-à-dire les paramètres qui seront testés lors de la cross validation. Nous utilisons un modèle avec une fonction de noyau kernel linéaire. De plus, nous spécifions une table de paramètre de la Cross-Validation (GridSearchCV()) qui reprend les hyper-paramètres et les paramètres de la segmentation évoqués plus haut. On cherche ensuite les paramètres les plus adaptés (best parameter) et d'après nos résultats les meilleurs paramètres seraient : 'C' : 10, 'max_iter' : 25000 dans le cadre d'un Kernel linéaire. On retient c'est paramètres car ce sont ceux ou le score de précision de la classification est le plus élevé en validation croisée.

Voici ci-joint les courbe de performance des différentes estimations :

FIGURE 13 – Taux de bien classé en CV en fonction du nombre d'iteration



Nous testons le meilleur modèle linéaire sur un échantillion test obtenu avec la fonction train test split.

La matrice de confusion du meilleur modèle est la suivante :

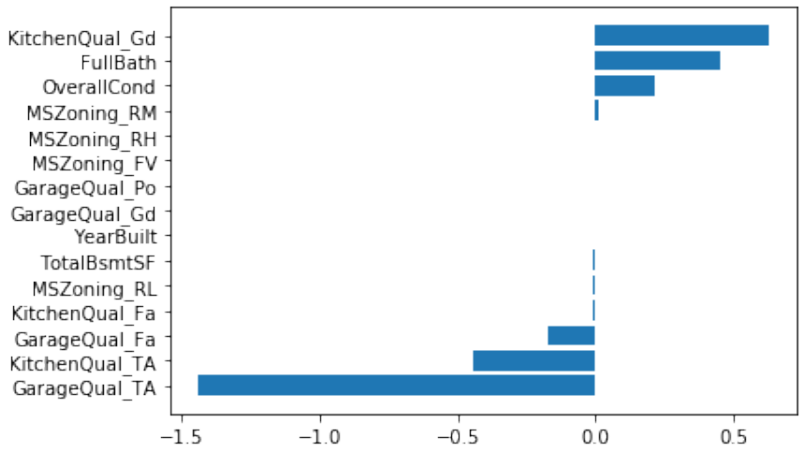
TABLE 2 – Matrice de confusion modèle SVM kernel linéaire

92	45	8
41	11	31
12	1	51

D'après le tableau on constate que le modèle prédit bien le fait d'appartenir à la classe 0 et plutôt bien le fait d'appartenir à la classe 2. Mais il prédit moyennement bien le fait d'appartenir à la classe 1. Le score de précision pour ce modèle est d'environ 51,36 %. Ce qui se rapproche du score de la CV.

Les coefficients les plus importants de ce modèle sont représentés dans le graphique ci-dessous :

FIGURE 14 – Coefficients des variables explicatives



Ainsi nous remarquons que la qualité de la cuisine (Gd = good), le nombre de salle de bains, et la qualité globale du bien ont de fort coefficients. On retrouve ainsi les variables qui semblait avoir une bon pouvoir prédictif sur les différentes classes avec l'analyse graphique.

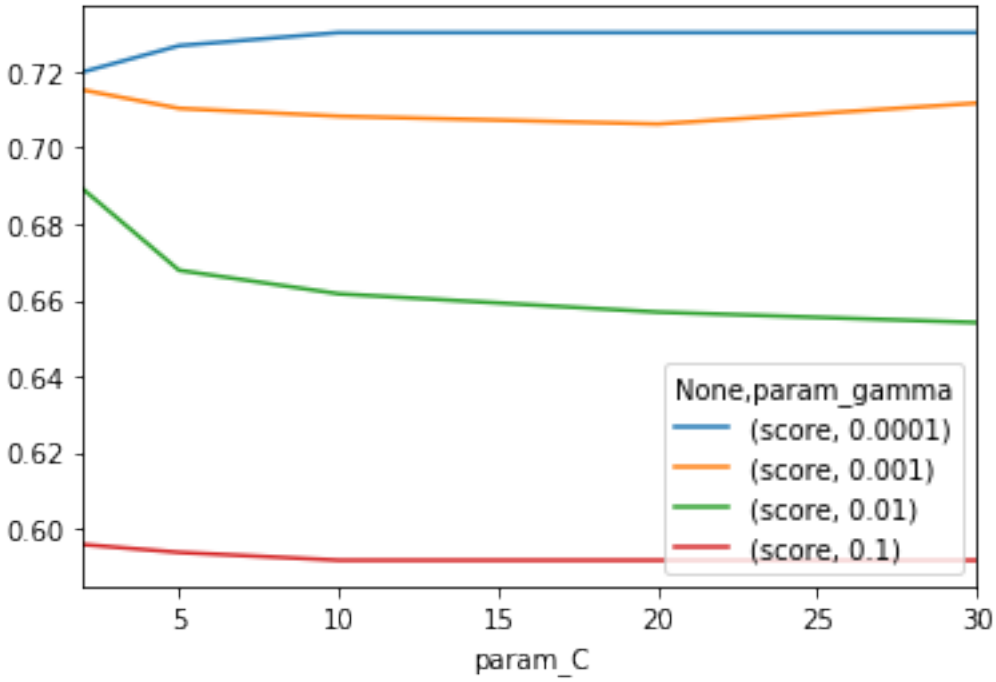
Ainsi, ce modèle n'est pas le meilleur possible puisqu'il prédit correctement que la moitié des observations lorsqu'il s'agit d'un échantillon inconnu. Ceci peut s'expliquer par le fait qu'il est impossible de prédire ce type de problème avec un séparateur linéaire. Pour cela nous allons essayer de modéliser des modèles SVM non linéaires (gaussien). Néanmoins on retrouve des variables avec un fort coefficients tel que la qualité de la cuisine, le nombre de salle de bain qui ressortaient lors de l'analyse graphique croisée.

4.1.2 *SVM avec kernel Gaussien*

Comme expliqué dans la première partie théorique, nous allons utiliser la méthode du noyau kernel. Avec cette méthode on souhaite trouver une frontière de décision non linéaire et donc peut être plus représentative. De la même manière que dans l'analyse du SVM linéaire, nous avons spécifié les différents hyper-paramètres C et Gamma qui seront testés par Cross Validation. On utilise toujours un modèle avec une fonction de noyau kernel linéaire et on spécifie une table de paramètre de la Cross-Validation. Les paramètres les plus adaptés (best paramètres) d'après nos résultats seraient : 'C' : 10, 'gamma' : 0.0001.

Voici ci-joint les courbe de performance des différentes estimations :

FIGURE 15 – Taux de bien classé en CV en fonction du paramètre C et γ



La matrice de confusion du meilleur modèle est la suivante :

TABLE 3 – Matrice de confusion modèle SVM kernel gaussien

126	12	7
25	52	6
6	18	40

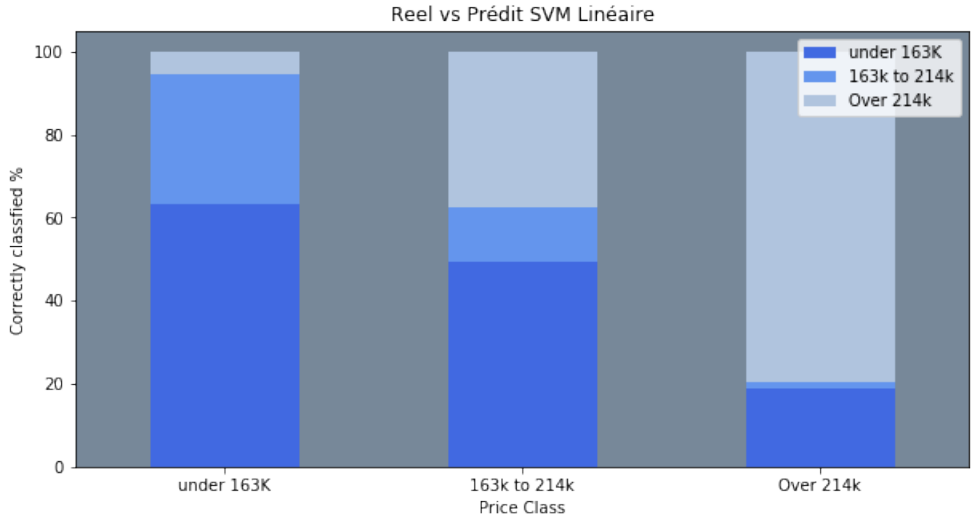
D'après le tableau on constate que le modèle prédit bien le fait d'appartenir à la classe 0, mais prédit moins bien le fait d'appartenir à la classe 1 et la classe 2. De plus, on note que la classe 2 est plus mal prédit avec ce modèlé qu'avec le modèlé linéaire. Le score de précision pour ce modèle est d'environ 74,65 %, ce qui témoigne d'une bonne qualité d'ajustement du modèle néanmoins. Les taux de faux positif et faux négatif sont plus bas qu'avec le modèle linéaire.

Dans les histogrammes en barres ci-dessous on peut visualiser facilement si le modèle prédit bien les différentes classes de prix ou non.

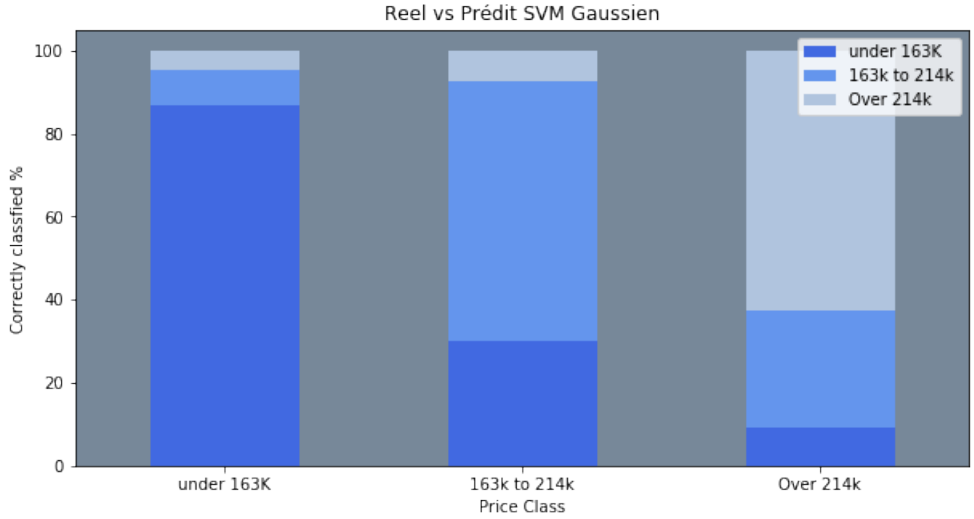
On constate ici que le SVM non linéaire (gaussien) prédisent mieux la classification des prix que le SVM linéaire.

FIGURE 16 – Comparaison modèle SVM linéaire & SVM gaussien

(a) SVM linéaire



(b) SVM gaussien



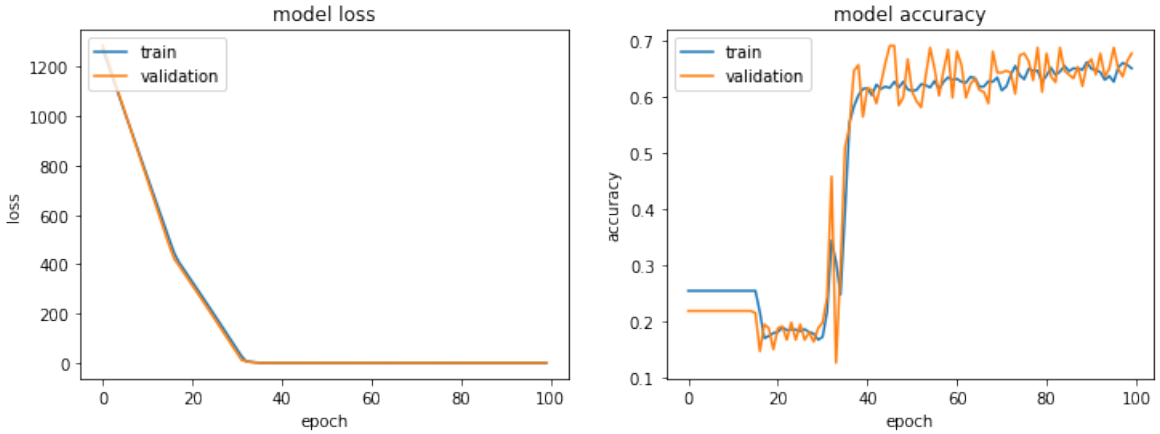
4.2 Modèles ANN

4.2.1 *ANN avec une couche cachée*

Les réseaux de neurones aussi sont des algorithmes très performants. Nous créons une fonction de perte pour les modèles ANN. Nous avons fixé le paramètre epoch à 100, i.e le nombre d'apprentissage que fait le modèle sur les données, afin d'éviter un surapprentissage. De plus on constate dans le graphique suivant que le score de précision est relativement stable (autour de 65% à partir de 70 epochs.

Pour un modèle ANN simple couche nous obtenons le graphique ci-dessous :

FIGURE 17 – Taux de bien classé en CV en fonction du nombre d'entrainement



La matrice de confusion est la suivante :

TABLE 4 – Matrice de confusion modèle ANN 1 couche cachée

On test ensuite le modèle ANN sur une échantillion test et nous obtenons les résultats suivants :

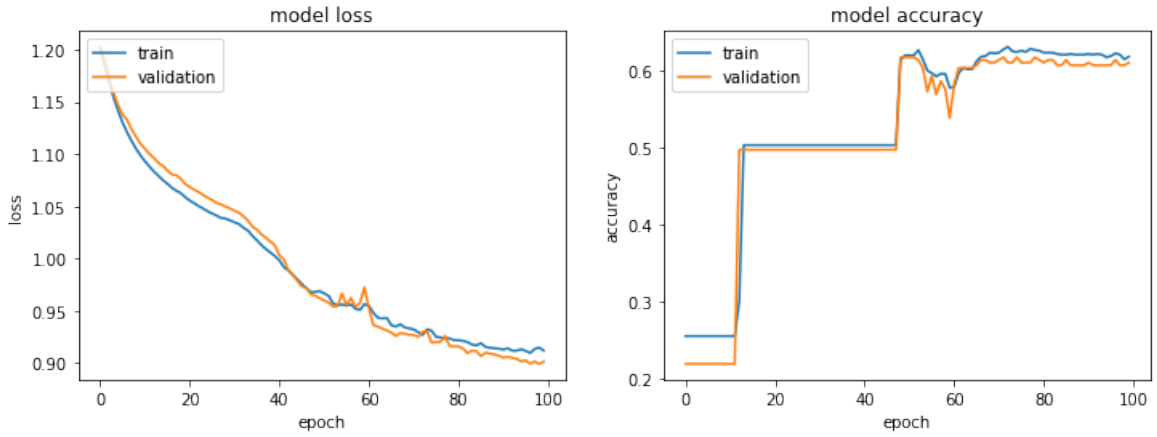
123	9	13
27	26	30
2	10	52

Comme pour les modèles précédents, on constate que le modèle prédit bien le fait d'appartenir à la classe 0, et 2 mais prédit moins bien le fait d'appartenir à la classe 1. Le score de précision pour ce modèle est d'environ 68,83 %, ce qui témoigne d'une qualité d'ajustement correcte pour ce modèle.

4.2.2 *ANN avec deux couches cachées*

Pour un modèle ANN avec deux couches cachées et 3 neurones,nous obtenons les résultats suivant en CV résumé dans le graphique ci-dessous :

FIGURE 18 – Taux de bien classé en CV en fonction du nombre d'entrainement



D'après le modèle ANN 2, on obteint un taux de bien classé en CV légèrement supérieur à 60 %, ce qui est inférieur au modèle ANN avec une seule couche caché.

Les résultats sur l'échantillon test sont les suivants :

TABLE 5 – Matrice de confusion modèle ANN 2 couches cachées

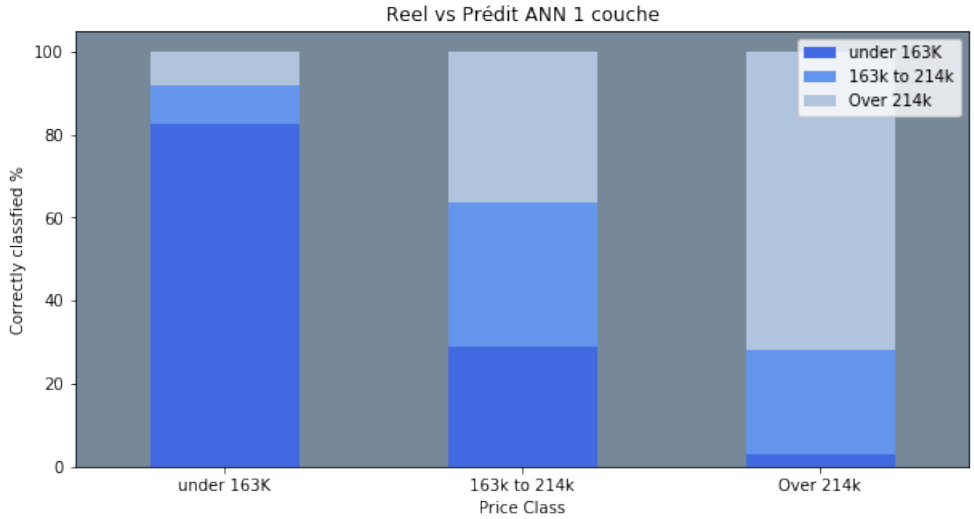
129	0	16
47	0	36
15	0	49

D'après le tableau on constate que le modèle prédit bien le fait d'appartenir à la classe 0, et la classe 2 mais prédit aucunement le fait d'appartenir à la classe 1. Le score de précision pour ce modèle est de 60,95 %, , ainsi notre taux de bien calssé est plus bas que celui du modèle ANN avec une seule couche cachée. Complexifier le modèle ici est plus délétaire par rapport à nos données.

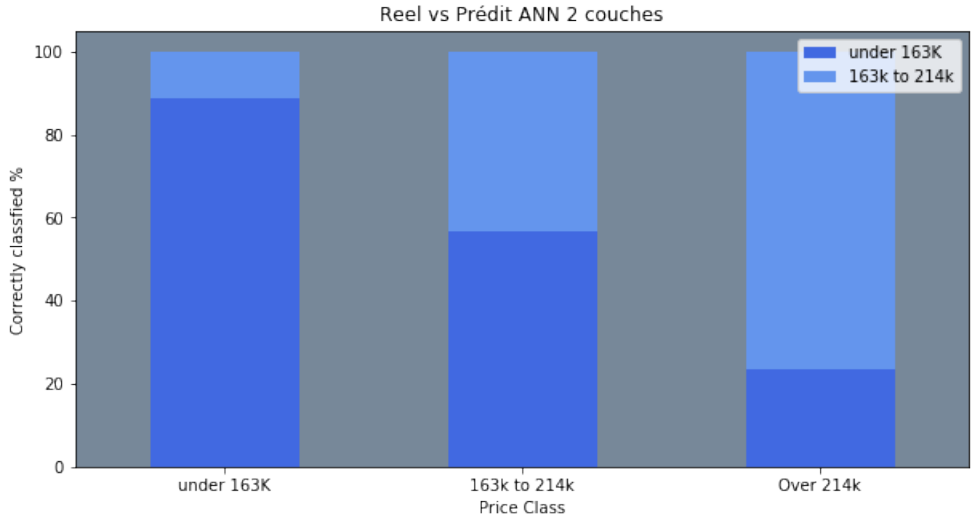
La figure suivante compare les réultats out of sample des deux modèles ANN.

FIGURE 19 – Comparaison modèle ANN 1 couche & ANN 2 couches

(a) ANN 1 couche



(b) ANN 2 couches



5 Conclusion

L'objectif de notre dossier était de modéliser différentes classes de prix en fonction de 8 variables explicatives tirées de la base de données de la compétition Kaggle. Nous avons estimé 4 modèles en tout : 2 SVM et 2 ANN. Il ressort comme principaux résultats que le modèle SVM avec kernel gaussien est le plus performant de tous en terme de précision de classement. On note par ailleur que la classe 1 et 2 sont les classes les moins bien prédites, et que la classe 1 est celle qui est la moins bien modélisé. On retrouve ces résultats pour tout les autres modèles. Enfin, avec le modèle SVM linéaire nous avons acceder aux coefficients des différentes variables. Comme préssenti dans l'analyse graphique bivarié,

nous retrouvons parmis les variables avec un fort coefficient : une qualité de cuisine élevé, le nombre de salle de bains augmentant la probabilité d'avoir une classe de prix élevé pour un bien donné. A contrario, une qualité de cuisine et de garage basse augmente la probabilité d'etre dans une classe de prix basse.

Table des figures

Liste des tableaux