Équipe 2

Travail fait par

Matis Brassard-Verrier (111 182 740)
Alyson Marquis (111 183 605)
Alexis Picard (111 182 200)
Samuel Provencher (111 181 794)

Apprentissage statistique en actuariat

ACT-3114

Rapport 2

Présenté à

Marie-Pier Côté

École d'actuariat
Université Laval
22 avril 2020

Table des matières

1 Introduction

Dans le cadre du travail, nous allons tenter de modéliser le prix de vente des maisons dans la région de Seattle (King County, USA) en utilisant de nombreuses caractéristiques ayant une incidence sur la valeur d'une maison. La variable réponse à prédire, soit le prix de vente d'une maison, est une valeur positive évaluée en dollars américains. La modélisation de cette variable pourrait être utile pour différentes raisons dans un contexte actuariel. Comme la somme assurée d'une maison a un lien très fortement proportionnel à son prix de vente, une compagnie d'assurance pourrait être intéressée de modéliser le prix de vente de maisons dans des nouveaux développements immobiliers afin de tenter de prédire les futures soumissions d'assurance habitation et d'offrir des offres personnalisées aux acheteurs de ces nouvelles maisons. De plus, dans un contexte de gestion des risques, certains assureurs ont un portefeuille de prêts hypothécaires ou utilisent des produits dérivés sur prêts hypothécaires pour se couvrir du risque (*hedging*). Ainsi, il pourrait être intéressant pour ces compagnies d'avoir une estimation des montants de prêts hypothécaires dans une région donnée en se basant sur le prix de vente des maisons afin de mieux gérer leur risque. La pertinence de tenter de prédire cette variable qu'est le prix de vente des maisons devient alors évidente.

Le jeu de données utilisé sera le suivant : **kc_house_sales (House sales in King County, USA)**. Dans les prochaines sections, sept modèles différents seront étudiés et comparés. Les deux modèles les plus performants seront également interprétés plus en profondeur. Pour construire nos modèles, nous utiliserons les données modifiées selon la méthodologie décrite dans la première partie du travail. Rappelons rapidement que certaines observations ont été supprimées du jeu de données et que nous avons créé manuellement trois nouvelles variables explicatives ; pour plus de détails, vous pouvez consulter la première partie du travail. 80 % des données seront utilisées pour effectuer l'entraînement des modèles et 20 % seront réservées pour tester les modèles et les comparer entre eux.

2 Modèle de base

Un bon modèle de base a été choisi en utilisant une technique étudiée dans le cours ACT-2003 Modèles linéaires en actuariat, soit la régression linéaire multiple. Ce type de modèle a été choisi en raison de sa simplicité et parce qu'il s'adapte bien au jeu de données. En effet, la variable réponse *price* est monétaire et possède une distribution asymétrique. Il a été vu qu'en présence de ce type de variable réponse, une régression linéaire multiple en appliquant une transformation logarithmique sur la variable réponse était appropriée. Tel que mentionné dans la première partie de ce travail, la transformation logarithmique permet de s'approcher de la distribution d'une loi normale, ce qui rend la variable réponse plus facile à modéliser. Pour construire le modèle, seul l'échantillon d'entraînement a été utilisé. De plus, le modèle utilise toutes les 17 variables explicatives. L'équation du modèle choisi est la suivante :

$$log(\hat{Y}_i) = \hat{\beta}_0 + \sum_{k=1}^{17} \hat{\beta}_k x_{i,k}$$

Ainsi, aucune interaction entre les variables explicatives n'a été considérée afin de garder le modèle simple et facilement interprétable. Certaines variables catégorielles à plusieurs niveaux, dont l'importance des interactions étaient négligeables, augmentaient le temps de calcul et rendaient le modèle plus difficilement interprétable. Ainsi, dans l'idée d'avoir un modèle de base simple, il a été décidé de ne pas considérer les interactions dans ce modèle. En outre, une sélection de variables formelles n'a pas été effectuée, contrairement à ce qui est habituellement fait lorsqu'on veut raffiner un modèle linéaire multiple.

3 Ajustement des modèles

La présente section présentera les six modèles testés. Il est à noter que, pour tous ces modèles, une transformation logarithmique a été effectuée sur la variable réponse *price*, tel qu'expliqué dans la section **2 - Modèle de base**. Cela nous permet de supposer une distribution normale de la variable réponse, et donc d'utiliser le critère de l'erreur quadratique moyenne (EQM) pour tous nos modèles pour valider leurs hyperparamètres et pour les comparer entre eux.

3.1 Modèle linéaire généralisé avec une régularisation Lasso

Dans le cadre du travail, il a été premièrement été choisi de construire un modèle linéaire généralisé avec une régularisation de type Lasso. Notre choix s'est arrêté sur ce type de régularisation, puisque la régularisation Lasso permet d'effectuer la sélection de variables. Pour ce faire, il suffit de minimiser l'équation de score suivante :

$$S^{Lasso} = \sum_{i=1}^{p}(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j|,$$

où p est le nombre de paramètres du modèle et λ est le paramètre de régularisation. La minimisation de cette équation pourra mener à des coefficients β exactement égaux à zéro, sélectionnant ainsi les variables du modèle. Nous avons utilisé la méthode implantée dans le paquetage **glmnet** pour choisir le paramètre λ ainsi que pour bâtir le modèle.

Afin de modéliser le prix de vente des maisons à King County, le modèle linéaire généralisé avec une régularisation Lasso a été construit à l'aide de l'échantillon d'entrainement. Le paramètre de régularisation a été choisi à l'aide d'une validation croisée à six plis. Cette validation croisée est intégrée dans la fonction `cv.glmnet`. Ainsi, la valeur optimale de ce paramètre était de $\lambda = 0.0000791264$. Le modèle retenu est composé de sept variables explicatives, soient *sqft_lot*, *waterfront*, *view*, *sqft_above*, *sqft_basement*, *lat* et *reno*. Le modèle est aussi constitué de 22 termes d'interaction. L'équation du modèle est la suivante :

$$log(\hat{Y}_i) = \hat{\beta}_0 + \sum_{k=1}^{7}\hat{\beta}_k x_{i,k} + \sum_{j=1}^{22}\hat{\alpha}_j z_{i,j} \quad ,$$

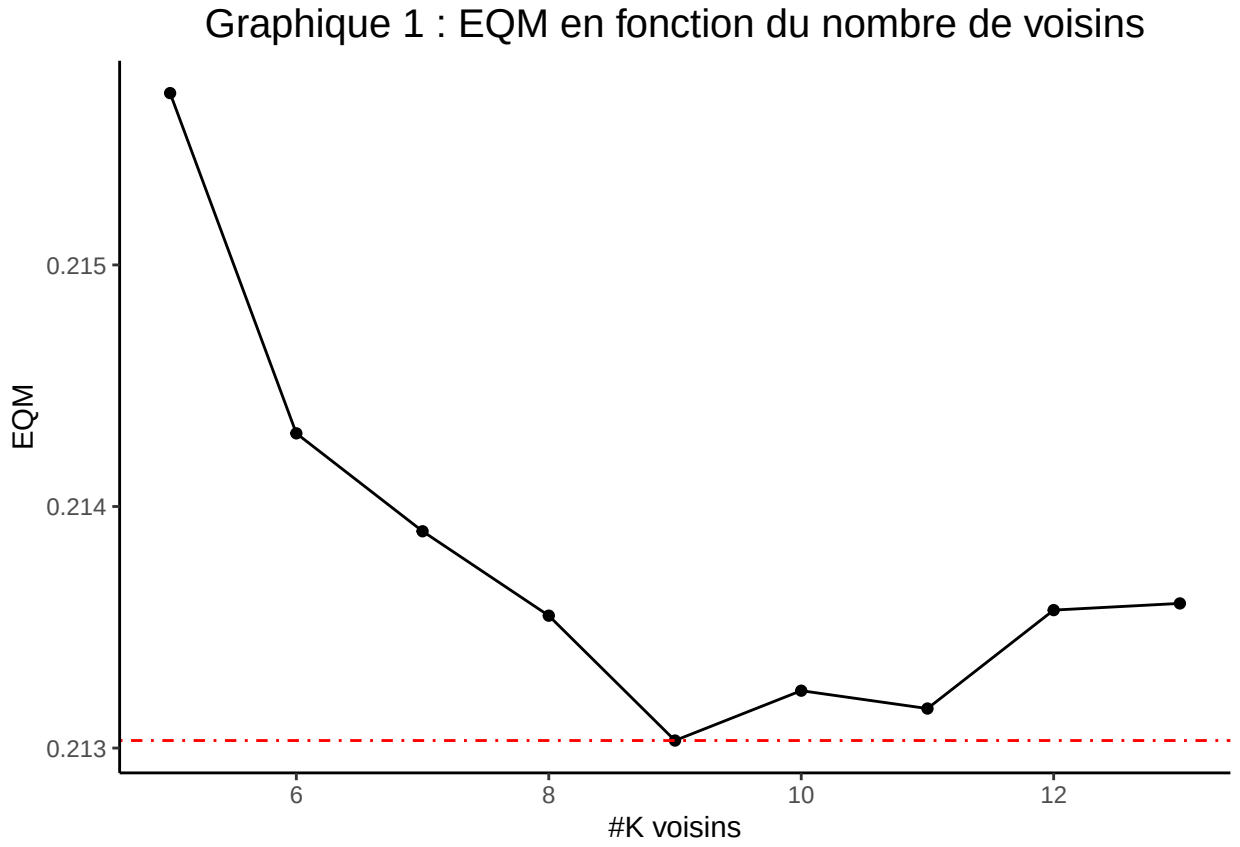
où $z_{i,j}$ est le $j^{ième}$ terme d'interaction.

3.2 Modèle des *k* plus proches voisins

Un autre modèle qui a été testé est celui des k plus proches voisins. Étant donné que nous sommes en présence d'un problème de régression, la fonction `knn.reg` du paquetage **FNN** a été utilisée pour construire ce modèle.

Le modèle des k plus proches voisins est simple. Afin d'effectuer une prédiction pour une observation dont les valeurs des variables explicatives sont comprises dans le vecteur $\mathbf{x}_0$, il faut regarder l'ensemble des k plus proches voisins de $\mathbf{x}_0$, c'est-à-dire les k observations qui en sont le plus près selon la distance euclidienne. Puis, la prévision du point $\mathbf{x}_0$ est la moyenne des variables réponses des observations comprises dans l'ensemble des k plus proches voisins.

Il faut d'abord déterminer la valeur optimale de k, soit le nombre de voisins à considérer. Pour ce faire, une validation croisée à dix plis a été utilisée. La fonction `train` du paquetage **caret** a été utilisée pour faire cette validation croisée. Étant donné que le modèle des k plus proches voisins est fondé sur la distance euclidienne, les données ont été standardisées avant de procéder à la validation croisée. La métrique choisie pour sélectionner la valeur de k est l'erreur quadratique moyenne (`metric="RMSE"`). Ainsi, la valeur de k qui minimisait l'erreur quadratique moyenne est $k = 9$, tel qu'illustré sur le graphique 1.



3.3 Arbre de décision

Un autre modèle qu'il a fallu tester est un arbre de décision, mais plus précisément dans le cas présent, un arbre de régression. Pour ce faire, l'algorithme *classification and regression tree (CART)* implanté dans le paquetage **rpart** a été utilisé.

Il a tout d'abord été décidé d'optimiser l'hyperparamètre `minbucket`, soit le nombre minimal d'observations dans une feuille de l'arbre. Une méthode manuelle a du être utilisée parce qu'aucune fonction intégrée connue ne permettait d'optimiser cet hyperparamètre. En premier lieu, l'échantillon d'entrainement a été séparé en échantillon de validation (20 %) et en un nouvel échantillon d'entrainement (80 %). L'échantillon de validation a été utilisé pour faire le choix optimal de l'hyperparamètre `minbucket`. En second lieu, plusieurs valeurs ont été testées entre `minbucket` $= 1$ et `minbucket` $= 200$ sur des arbres élagués afin de se donner une idée non raffinée. Par la suite, la recherche a été raffinée et la valeur optimale ainsi trouvée était de `minbucket` $= 7$. Cette valeur était celle qui minimisait l'erreur quadratique moyenne des arbres élagués sur l'échantillon de validation.

Une fois cet hyperparamètre défini, un arbre de régression a été construit en utilisant toutes les variables explicatives ainsi qu'un paramètre de complexité nul ($cp = 0$). L'arbre de régression a été obtenu en spécifiant `method="anova"`, ce qui a permis de trouver l'arbre minimisant l'erreur quadratique moyenne. Seul l'échantillon d'entraînement initial a été utilisé pour entraîner ce modèle, l'échantillon test étant réservé pour analyser les performances prédictives du modèle. Afin d'optimiser le paramètre de complexité, une validation croisée à dix plis a été effectuée. Cette validation croisée est implantée de base dans la fonction `rpart`, donc aucune programmation supplémentaire n'a été nécessaire. Ainsi, le paramètre de complexité optimal obtenu était de 0.0000734972. Ce choix optimal a été utilisé pour élaguer l'arbre de régression et ainsi réduire la variance de la prédiction. L'arbre élagué représente donc le meilleur compromis entre le biais et la variance de la prédiction pour ce type d'arbre.

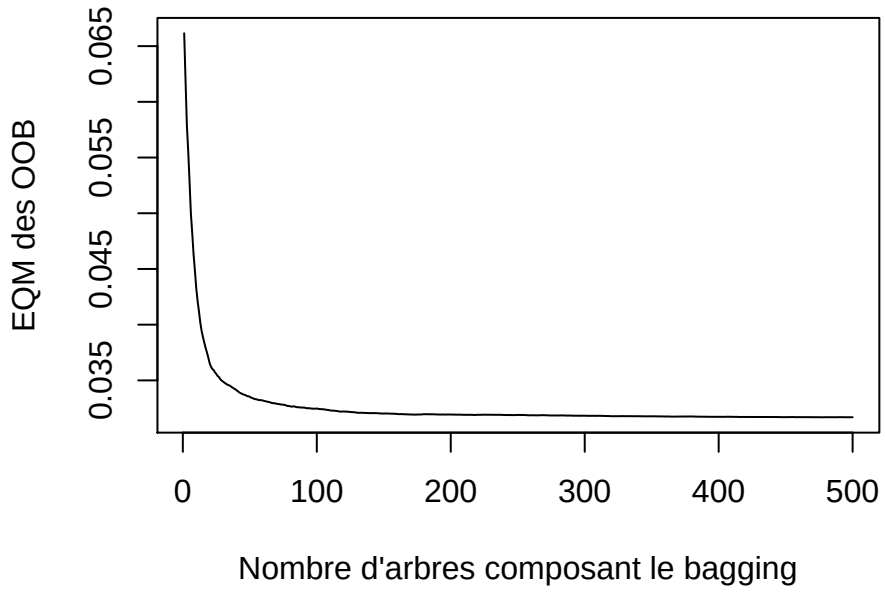
Il aurait été intéressant de représenter graphiquement ce modèle puisqu'il s'agit d'une façon de bien comprendre ce type de modèle. Cependant, il n'a pas été possible de présenter ce modèle sous forme de graphique. En effet, malgré le fait que l'arbre ait été élagué, il était trop complexe pour être agréable à l'oeil. Ceci est causé par le fait que le modèle a été entrainé avec 13821 observations, ce qui est un nombre assez important.

3.4 Ensemble d'arbres de décisions agrégés par *bagging*

Un autre modèle qui a été testé est celui du *bagging* avec des arbres de régression. Étant donné le type de la variable réponse recherchée, soit le prix d'une maison, des arbres de régression ont été choisis pour effectuer le *bagging*. Pour ce faire, l'algorithme *classification and regression with Random Forest (randomForest)* implanté dans le paquetage **randomForest** a été utilisé.

Un échantillon *bootstrap* avec remise de la même taille que l'échantillon d'entrainement a été sélectionné par l'algorithme `randomForest` (`sampsize= nrow(donnees.train)`). Le nombre de variables échantillonnées aléatoirement a donc été déterminé à 17 (`mtry=17`) étant donné que la base de données d'entrainement comportait 18 variables, dont la variable réponse. Le seul hyperparamètre qui était à déterminer était celui du nombre d'arbres de régression créés avec le bagging. Pour ce faire, un graphique de l'erreur quadratique moyenne (EQM) des observations *out-of-bag* (OOB) en fonction du nombre d'arbres composant le bagging a été tracé afin de déterminer quand l'EQM des OOB se stabilisait. Sur le graphique 2, on voit que 500 arbres étaient suffisants et c'est donc le nombre qui a été retenu (`ntree=500`).

Graphique 2 : EQM des observations OOB en fonction du nombre d'arbres



À noter qu'aucun élagage n'a été fait sur les arbres de régression créés par le bagging, donc ceux-ci avaient un paramètre de complexité `cp=0`.

3.5 Forêt aléatoire

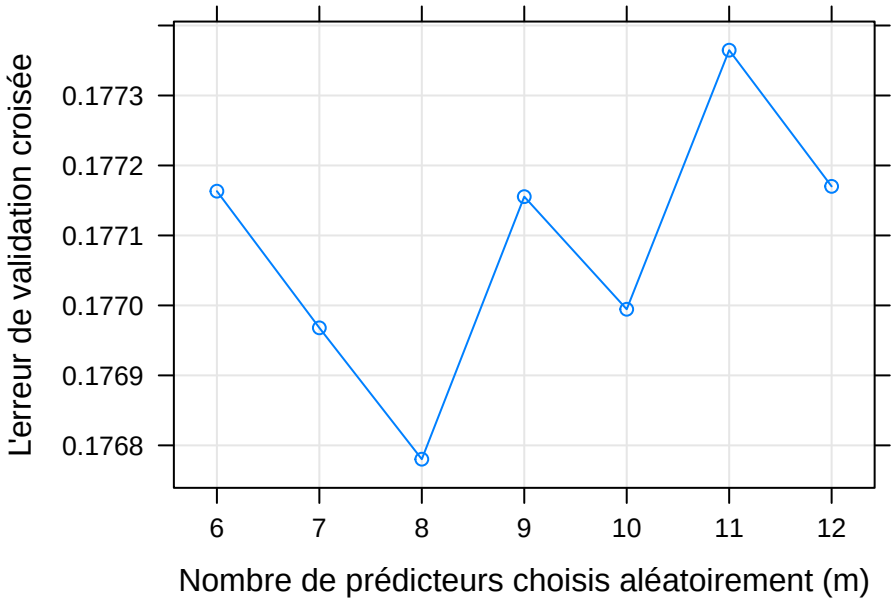
Un autre modèle qui a été testé est celui de la forêt aléatoire. Pour ce faire, le même algorithme *classification and regression with Random Forest (randomForest)* implanté dans le paquetage **randomForest** a été utilisé.

Le *bagging* et la forêt aléatoire sont très similaires, mais la forêt aléatoire permet de décorreler les arbres à l'aide de deux méthodes.

Tout d'abord, on choisit un échantillon *bootstrap* plus petit que le *bagging*, c'est pourquoi dans notre cas un `sampsize = 0.75*nrow(donnees.train)` a été choisi, ce qui équivaut à utiliser 75% des données de l'échantillon d'entrainement.

Puis, à chaque itération dans la construction de l'arbre, on choisit aléatoirement m prédicteurs qui seront les candidats pour la séparation (`mtry`). Ce choix de m optimal a été fait à l'aide d'une validation croisée à cinq plis. La fonction `train` du paquetage **caret** a été utilisée pour faire cette validation croisée. La métrique choisie pour sélectionner la valeur de m est l'erreur quadratique moyenne (`metric="RMSE"`). Ainsi, la valeur de m qui minimisait l'erreur quadratique moyenne de validation croisée est $m = 8$, comme le montre le graphique 3 :

Graphique 3 : EQM des observations OOB en fonction du mtry



Pour ce qui est des autres hyperparamètres, hormis le nombre d'arbres (`ntree`) qui a été abaissé à 150 pour des fins d'optimisation et de vitesse de calcul, les mêmes valeurs ont été gardées pour le *bagging* et la forêt aléatoire.

3.6 Modèle de boosting de gradient stochastique

Enfin, le dernier modèle ajusté aux données en est un de boosting de gradient stochastique. Nous allons encore une fois supposer que la variable réponse suit une distribution gaussienne, ce qui nous permet d'utiliser l'erreur quadratique moyenne (EQM) comme fonction de perte pour construire le modèle.

En quelques mots, le modèle de boosting de gradient stochastique est une procédure itérative : à chaque itération, un arbre de régression est ajusté aux gradients négatifs de la fonction de perte de l'itération précédente. La prévision de chaque itération est prise en compte dans le modèle proportionnellement à un paramètre λ appelé paramètre de régularisation. Plus ce λ est petit, plus le modèle apprend petit à petit en n'accordant pas trop d'importance à chaque prévision et plus la performance finale du modèle sera bonne, mais il faudra cependant plus d'itérations pour obtenir un modèle final optimal. Il faut également faire attention de ne pas construire un modèle avec trop d'arbres, car les modèles de boosting de gradient stochastique ont un risque de surajustement.

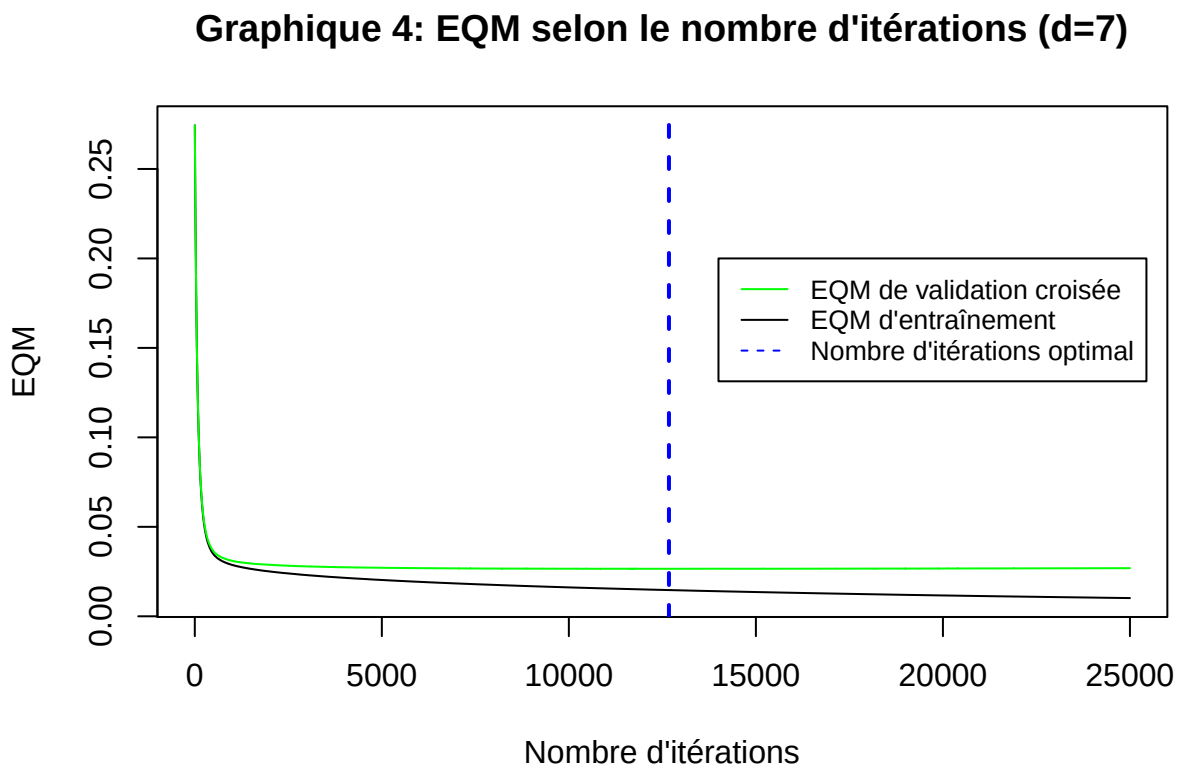
Nous avons choisi un paramètre de régularisation λ de 1 %. Ce paramètre nous a permis d'obtenir un

compromis efficace entre temps de calcul et précision du modèle.

Nous avons utilisé les fonctions intégrées au paquetage **gbm** pour procéder à l'optimisation et à la construction du modèle. En utilisant une validation croisée à cinq plis, nous avons optimisé les paramètres d (la profondeur maximale de l'arbre ajusté à chaque itération) et T (le nombre d'itérations du modèle). Des valeurs de trois, cinq, sept et neuf ont été testées pour la profondeur des arbres d.

Pour ce qui est des autres hyperparamètres, soit le pourcentage de sous-échantillonnage δ et le nombre minimal d'observations dans chaque noeud pour les arbres, nous avons gardé leurs valeurs par défaut, soit respectivement 50 % de sous-échantillonnage et dix observations. En effet, étant donné le λ choisi et la taille de notre jeu de données, nous avons déterminé qu'optimiser ces hyperparamètres nécessiterait un temps de calcul excessif pour les avantages qu'il serait possible d'en retirer.

À titre d'exemple, voici le graphique d'optimisation permettant de déterminer le nombre d'itérations optimal pour le modèle utilisant des arbres de profondeur sept :



La même méthode a été utilisée pour déterminer le nombre d'itérations optimal pour chaque paramètre d testé.

Pour déterminer lequel de ces modèles est le meilleur, nous les avons comparés selon le critère de l'erreur quadratique moyenne minimale de validation croisée à cinq plis.

Table 1: L'EQM des modèles testés

Profondeur d	Nombre d'itérations optimal	EQM de validation croisée
3	29884	0.027029
5	18905	0.026559
7	12676	0.026556
9	10515	0.026378

Selon ce critère, le meilleur modèle est celui obtenu avec $d = 9$ et 10515 itérations. C'est donc ce modèle que nous utiliserons lors de la comparaison des différents modèles entre eux.

4 Comparaison des modèles

La métrique utilisée pour comparer la puissance prédictive des différents modèles est l'erreur quadratique moyenne (EQM). Cette métrique a été choisie puisque nous sommes en présence d'un problème de régression et que nous supposons une distribution normale pour la variable réponse. De plus, les autres métriques vues dans le cadre du cours ACT-3114 Apprentissage statistique en actuariat sont utilisées pour des problèmes de classification ou lorsque la variable réponse est supposée suivre une loi Poisson. Ainsi, l'EQM est la métrique la plus appropriée pour notre problème. Pour chacun des modèles, l'EQM a été calculée avec les données de test afin de ne pas utiliser les mêmes données qui avaient déjà été utilisées pour entrainer les modèles.

Table 2: L'EQM des sept modèles testés

Modèle	EQM
Modèle de base	0.05300
Modèle Lasso	0.05245
Modèle des k plus proches voisins	0.04570
Arbre de décision	0.04962
Bagging	0.03350
Forêt aléatoire	0.03253
Boosting de gradient stochastique	0.02717

Les valeurs de l'EQM pour les sept modèles testés sont présentées dans le Tableau 1. Les deux meilleurs modèles selon cette métrique sont le modèle de forêt aléatoire et le modèle de boosting de gradient stochastique. En effet, leur EQM est respectivement de 0,03253 et de 0,02717.

Dans la prochaine section, les résultats obtenus avec le meilleur modèle seront brièvement analysés et visualisés. Les deux meilleurs modèles seront également interprétés plus en détail.

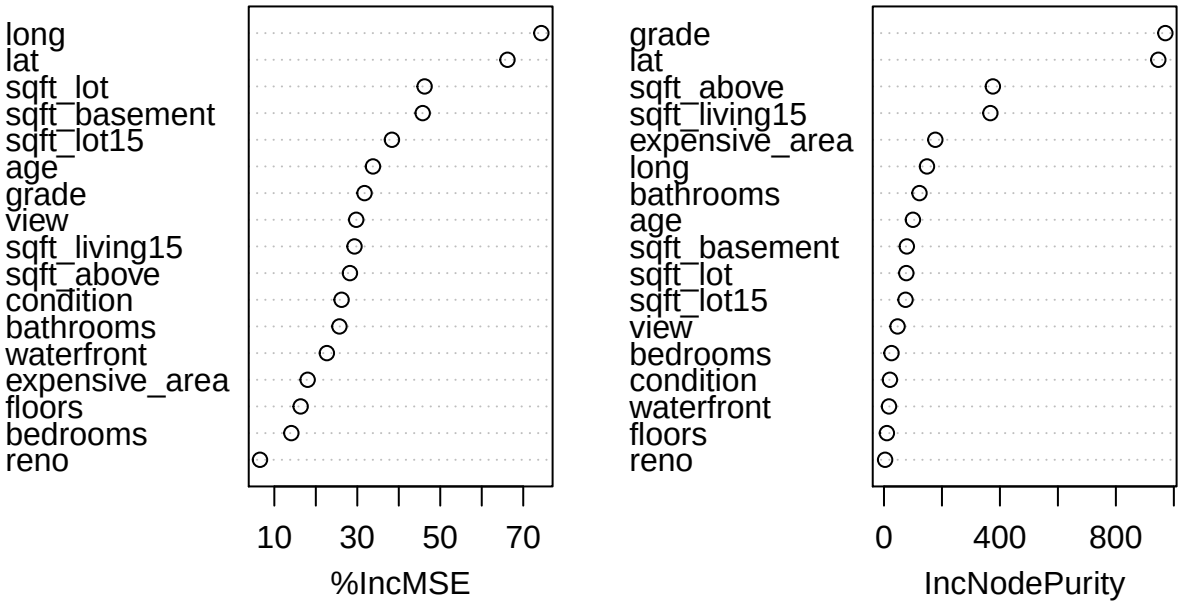
5 Interprétation des meilleurs modèles

5.1 Forêt aléatoire

Notre deuxième meilleur modèle obtenu en termes de prévision est celui de la forêt aléatoire. Ce modèle est difficilement interprétable étant donné le mélange de plusieurs arbres de régression différents. Voici toutefois ce que le modèle peut nous dire en effectuant une analyse approfondie.

Si l'on affiche l'importance des variables faisant partie de notre jeu de données en fonction de deux mesures différentes, soit la *mean decrease in accuracy* et la *mean decrease in node impurity*, on obtient le graphique ci-dessous :

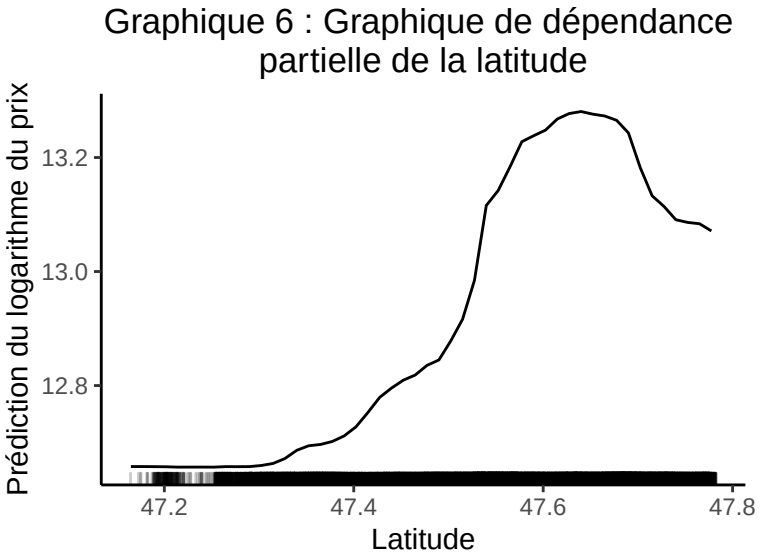
Graphique 5 : Importance des variables composant la forêt aléatoire



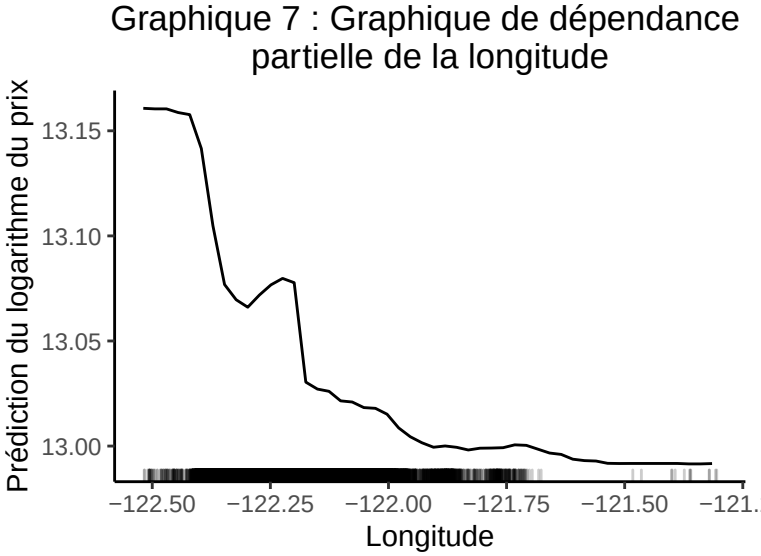
Si on permute de façon aléatoire les valeurs de la variable à analyser voulue, on calcule de nouvelles prévisions sur nos données et ensuite on compare la nouvelle erreur quadratique moyenne avec celle du jeu de données initial, on obtient la *mean decrease in accuracy*. Pour notre jeu de données, on voit donc que les variables de la latitude (*lat*) et la longitude (*long*) sont celles qui ont le plus d'incidence sur nos prévisions du logarithme de prix d'achat de la maison si on applique cette permutation.

Si on calcule la diminution moyenne dans l'erreur quadratique moyenne due à une séparation sur la variable à analyser voulue, on obtient la *mean decrease in node impurity*. Avec cette mesure, ce ne sont pas les deux mêmes variables qui sont dites les plus importantes pour notre modèle. En fait, la latitude (*lat*) est toujours d'une grande importance, mais la qualité de la construction et de la conception (*grade*) apparait comme étant celle qui se démarque aussi.

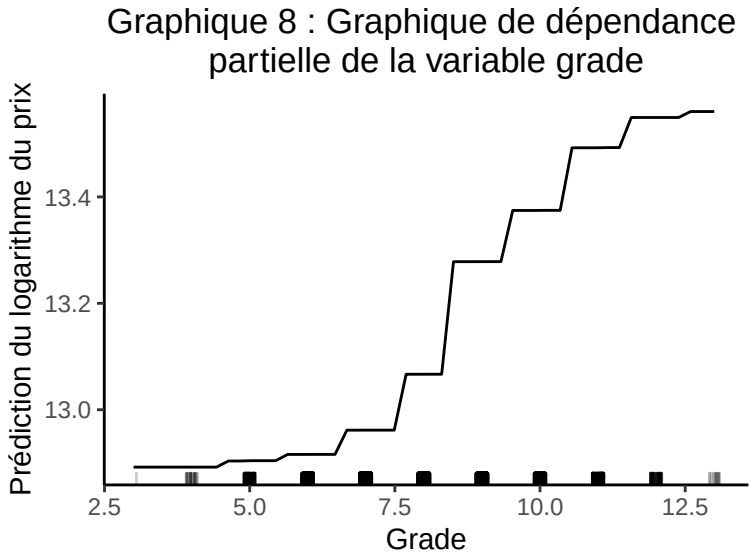
Pour mieux comprendre l'effet marginal de ces trois variables explicatives sur la prévision, on peut regarder leurs graphiques de dépendance partielle (PDP) :



Sur le graphique 6, on voit clairement que lorsque la latitude augmente, le logarithme du prix prédit de la maison augmente aussi. Cette tendance est véridique jusqu'à une latitude d'environ 47,7. Le sommet de la courbe est atteint entre les latitudes de 47,6 et de 47,7, ce qui est attendu puisqu'il s'agit de l'endroit approximatif du centre-ville. En fait, puisque la latitude exprime la position d'une maison selon l'axe nord-sud, on peut déduire qu'une maison située plus au nord de la région est plus dispendieuse qu'une maison située au Sud. Le graphique de la latitude concorde d'ailleurs avec la carte thermique de toutes les maisons vendues à King County du premier rapport où l'on voit que Seattle, la capitale, est située environ entre les parallèles 47,6 et 47,7.

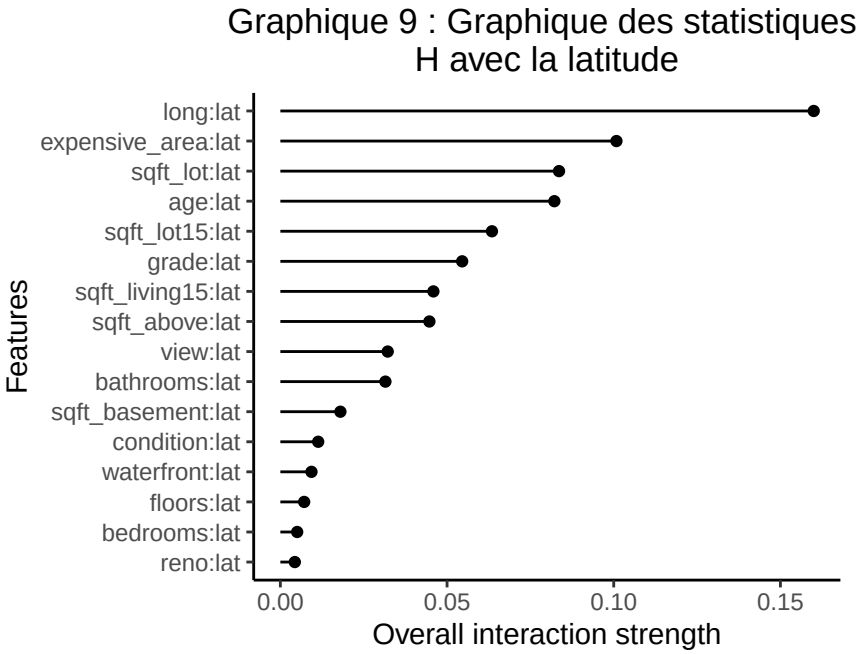


Sur le graphique 7, on voit clairement que lorsque la longitude augmente, le logarithme du prix de la maison diminue. En fait, puisque la longitude exprime la position d'une maison selon l'axe est-ouest, on peut déduire qu'une maison située à l'ouest de la région évaluée est plus dispendieuse qu'une maison située à l'est. Le graphique de la longitude concorde d'ailleurs également avec la carte thermique de toutes les maisons vendues à King County où l'on voit que les maisons dispendieuses, dont celles près du centre-ville, sont situées à l'ouest.



Sur le graphique 8, on voit clairement que lorsque la qualité de la construction et de la conception (*grade*) augmente, le logarithme du prix de la maison augmente aussi. En fait, de manière logique, si une maison à une meilleure qualité de construction et de conception, elle aura tendance à valoir plus cher. On peut déduire qu'une maison avec un *grade* élevé est plus dispendieuse qu'une maison avec un *grade* faible. On note également que l'augmentation dans le logarithme du prix de la maison apparait surtout à partir d'un *grade* de 5 ou supérieur.

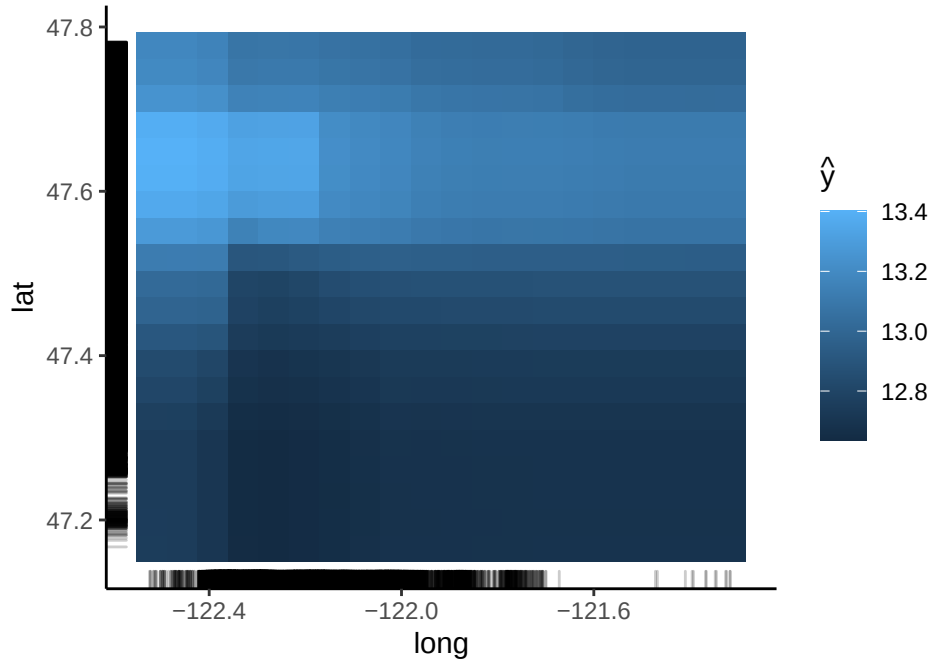
La variable *lat* étant importante selon les deux méthodes de mesure d'importance, il est donc intéressant d'aller voir si elle interagit avec d'autres variables. Une manière de quantifier ces interactions est de déterminer les statistiques H de Friedman : plus ces dernières sont grandes, plus l'interaction est forte. Ces statistiques H ont été déterminées à l'aide du paquetage **iml**. Le graphique des statistiques H de Friedman est présenté ci-dessous :



Il est intéressant de noter l'interaction entre la latitude et la variable *expensive_area*. En effet, la variable

expensive_area a été créée dans la première partie du présent travail. L'interaction est donc logique, car la variable *expensive_area* dépend de la latitude. Toutefois, les variables de la latitude et de la longitude sont celles qui intéragissent le plus ensemble par rapport aux autres interactions possibles. C'est d'ailleurs les deux variables les plus importantes selon la méthode *mean decrease in accuracy*. Le graphique de dépendance partielle bivarié entre ces deux variables serait donc fort intéressant. Il est présenté ci-dessous :

Graphique 10 : Graphique de dépendance partielle bivarié entre la latitude et la longitude



Ce graphique est très intéressant puisqu'il ressemble fortement à la carte thermique de tous les prix des maisons dans la région de King County présentée dans la première partie du travail. En effet, la majorité des maisons plus dispendieuses se situent au nord-ouest de la région comme le montre le coin bleu pâle sur le graphique ci-dessus. On observe aussi la forte interaction : en effet, les logarithmes de prix prédit varient beaucoup moins avec la longitude pour les latitudes faibles que pour les latitudes élevées.

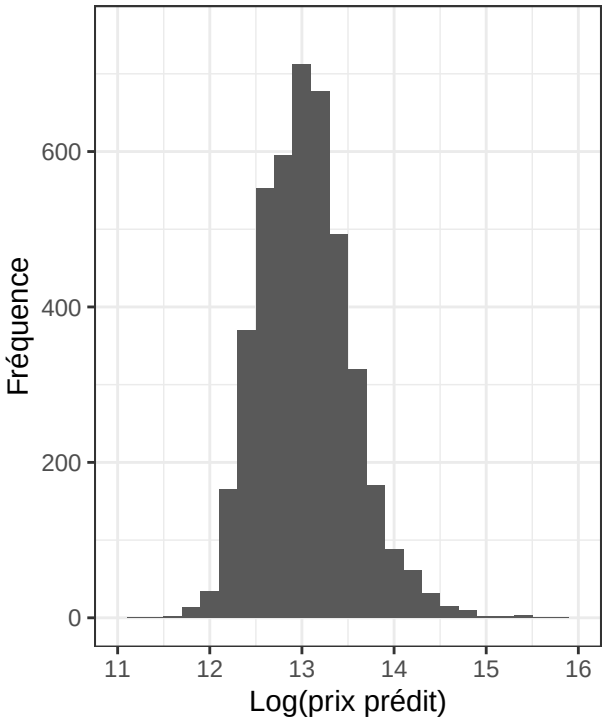
L'avantage de la forêt aléatoire est que l'utilisation d'un nombre élevé d'arbres permet de réduire la variance des prévisions tout en gardant un biais faible. De plus, elle permet de décorroler les arbres, ce qui constitue une amélioration par rapport au *bagging*. Ainsi, la forêt aléatoire arrive à prédire de très bonnes prévisions pour la variable réponse, comme en témoigne son EQM faible. Cependant, d'un point de vue technique, ces arbres sont lourds et très nombreux. La construction de la forêt aléatoire est donc très long. Ce problème de calcul est amplifié lorsque la base de données d'entrainement est grande comme c'est le cas pour notre problème.

5.2 Boosting de gradient stochastique

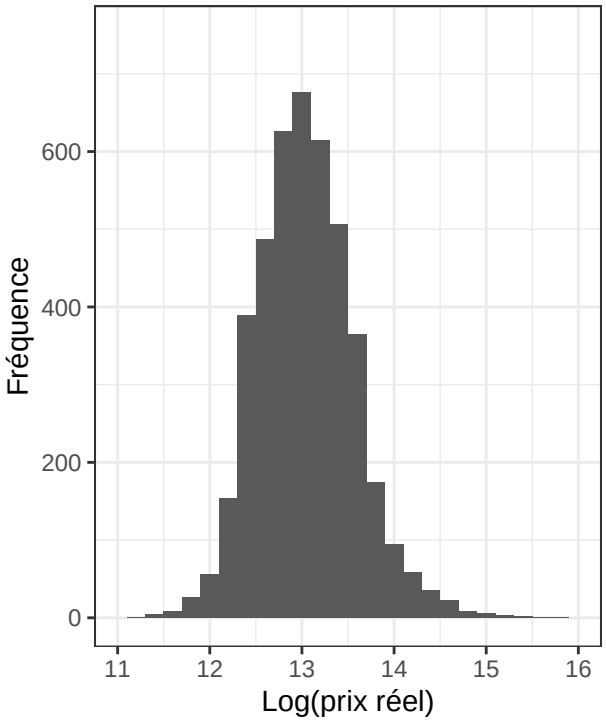
5.2.1 Analyse des résultats obtenus avec le modèle

Nous allons maintenant analyser très brièvement les résultats obtenus sur les données de l'échantillon de test en utilisant notre meilleur modèle, soit celui de boosting de gradient stochastique. Pour ce faire, nous allons comparer l'histogramme des logarithmes de prix prédits par le GBM sur l'échantillon de test à celui des logarithmes de prix réels.

Graphique 11 : Histogramme des logarithmes de prix prédits par le GBM



Graphique 12 : Histogramme des logarithmes de prix réels



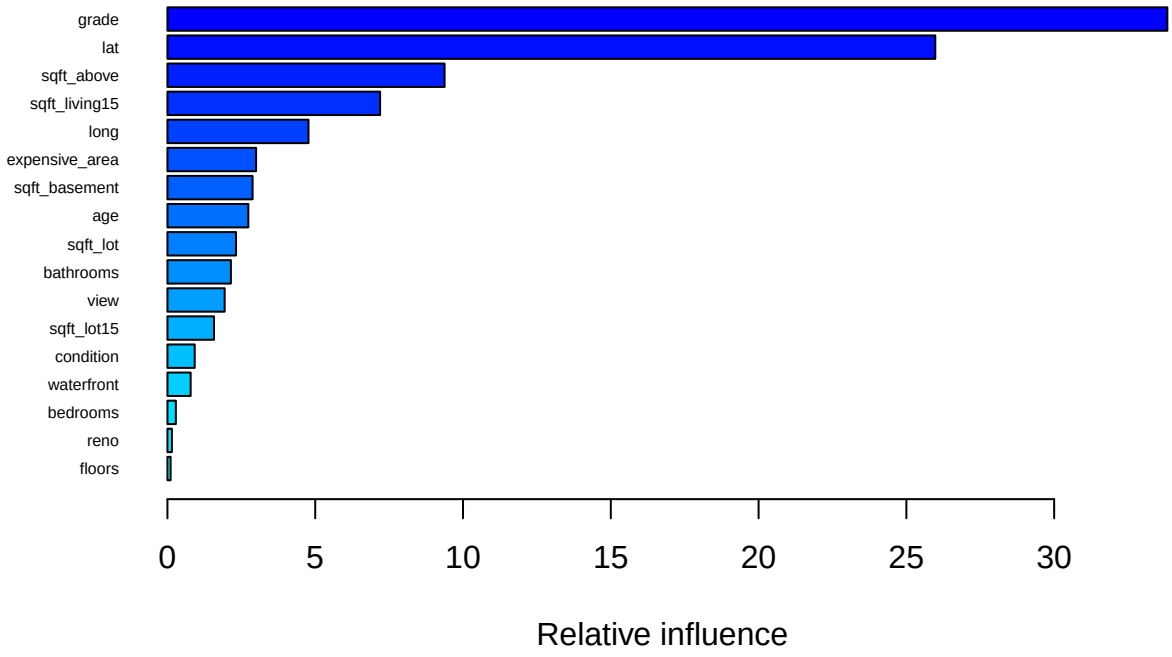
Nous observons que les logarithmes de prix prédits semblent suivre une distribution très similaire à celle des prix réels, ce qui témoigne à nouveau de la force de notre modèle. En effet, les deux semblent suivre une distribution normale centrée autour d'une moyenne d'environ 13. Cependant, on peut remarquer que le kurtosis de la distribution des logarithmes de prix prédits semble légèrement plus élevé que celui de la distribution des logarithmes des prix réels ; en d'autres termes, les prédictions du GBM semblent légèrement plus centrées autour de la moyenne que ce qu'on observe avec les prix réels. Nous en concluons que notre modèle a un peu de difficulté à prévoir précisément les maisons de prix très faible ou très élevé, et que le modèle prédira des prix trop près de la moyenne pour ces maisons. La performance globale du moyenne est toutefois très bonne.

5.2.2 Interprétation du modèle

Enchaînons avec l'interprétation du modèle de boosting de gradient stochastique. Ce type de modèle est reconnu comme étant difficilement interprétable intuitivement, mais divers outils permettent d'ouvrir la boîte noire du modèle et d'y voir plus clair. Nous utiliserons les fonctions intégrées au paquetage **gbm** pour interpréter le modèle.

Tout d'abord, nous pouvons déterminer l'importance des différentes variables dans le modèle selon le critère de diminution moyenne de l'EQM causée par chaque variable dans les arbres.

Graphique 13 : Importance des variables dans le GBM

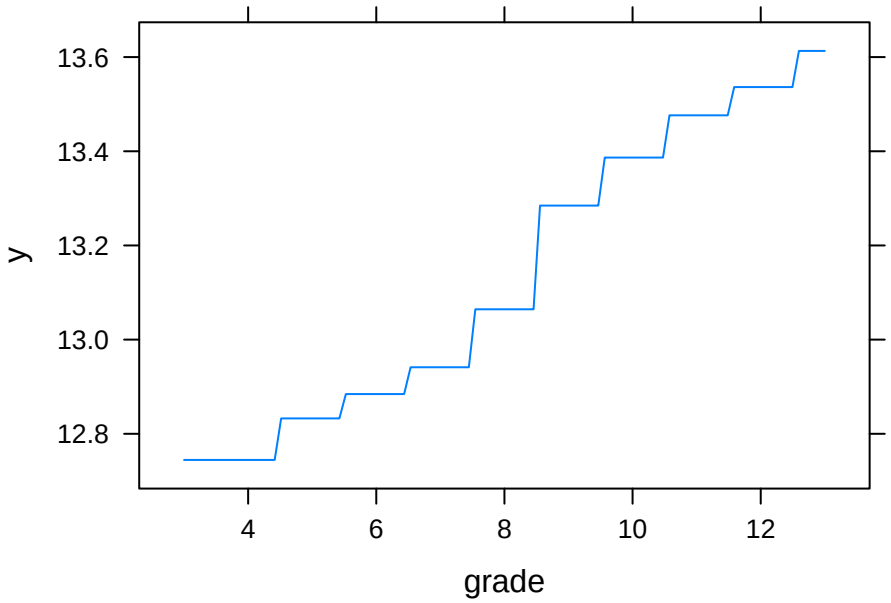


Nous observons que les variables les plus utiles dans le modèle de boosting sont *grade*, *lat*, *sqft_above*, *sqft_living15* et *long*. De plus, ces cinq variables les plus importantes sont les mêmes que celles du modèle de forêt aléatoire selon le même critère.

Une bonne manière d'examiner plus en détail l'incidence de chaque variable dans le modèle est de tracer des graphiques de dépendance partielle (PDP). Nous allons présenter les PDP des trois variables les plus importantes du modèle.

Commençons avec le PDP de la variable *grade*, soit l'indice de qualité de la construction et de la conception des maisons.

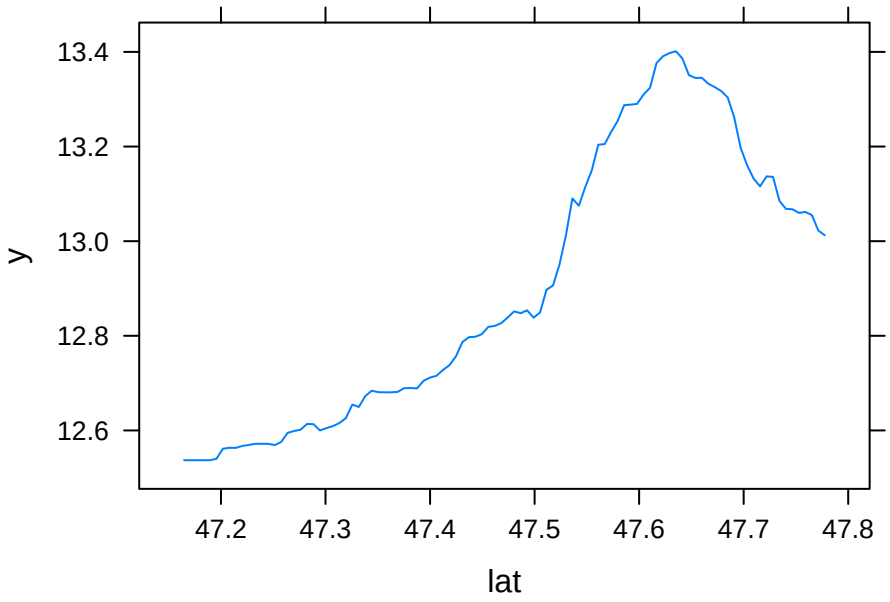
Graphique 14 : graphique de dépendance partielle de la variable grade



L'effet marginal de la variable *grade* est très semblable à celui observé plus tôt avec le modèle de forêt aléatoire : le logarithme du prix des maisons prédit est une fonction monotone croissante en fonction de la qualité de construction des maisons. Encore une fois, il est logique qu'une maison bien construite et conçue vaille plus cher qu'une autre maison moins bien construite, *ceteris paribus*. La différence entre les logarithmes des prix moyens des maisons les mieux construites et les moins bien construites est d'environ 0,7. Si on quitte l'échelle logarithmique, cela correspond à une augmentation d'environ 101,38 % du prix moyen entre les maisons les moins bien construites et les mieux construites. Autrement dit, on peut s'attendre en moyenne à ce que le prix d'une maison extrêmement bien construite soit environ le double de celui d'une maison extrêmement mal construite. On voit donc que la variable *grade* a un caractère très important sur le prix des maisons, en plus de suivre une relation proportionnelle bien définie avec le prix, ce qui explique la grande force de prédiction de cette variable.

Enchaînons avec le PDP de la variable *lat*, soit la latitude des maisons.

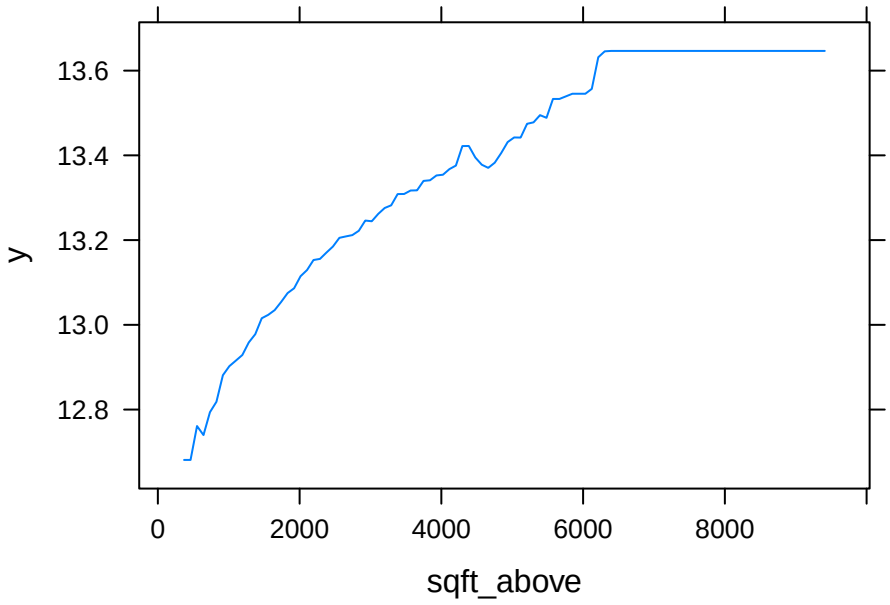
Graphique 15 : graphique de dépendance partielle de la variable lat



Encore une fois, le PDP est très similaire à celui obtenu en utilisant le modèle de forêt aléatoire. Le PDP obtenu avec le modèle de boosting de gradient stochastique semble cependant osciller plus, donc il a plus de variance et est plus ajusté aux données utilisées. Comme précédemment, on voit que le logarithme moyen du prix prédit est plus faible dans les zones de latitude faible (zone rurale au sud du comté), puis monte beaucoup à l'approche de la zone urbaine de Seattle pour enfin redescendre lorsqu'on atteint une nouvelle zone moins urbaine au nord du comté.

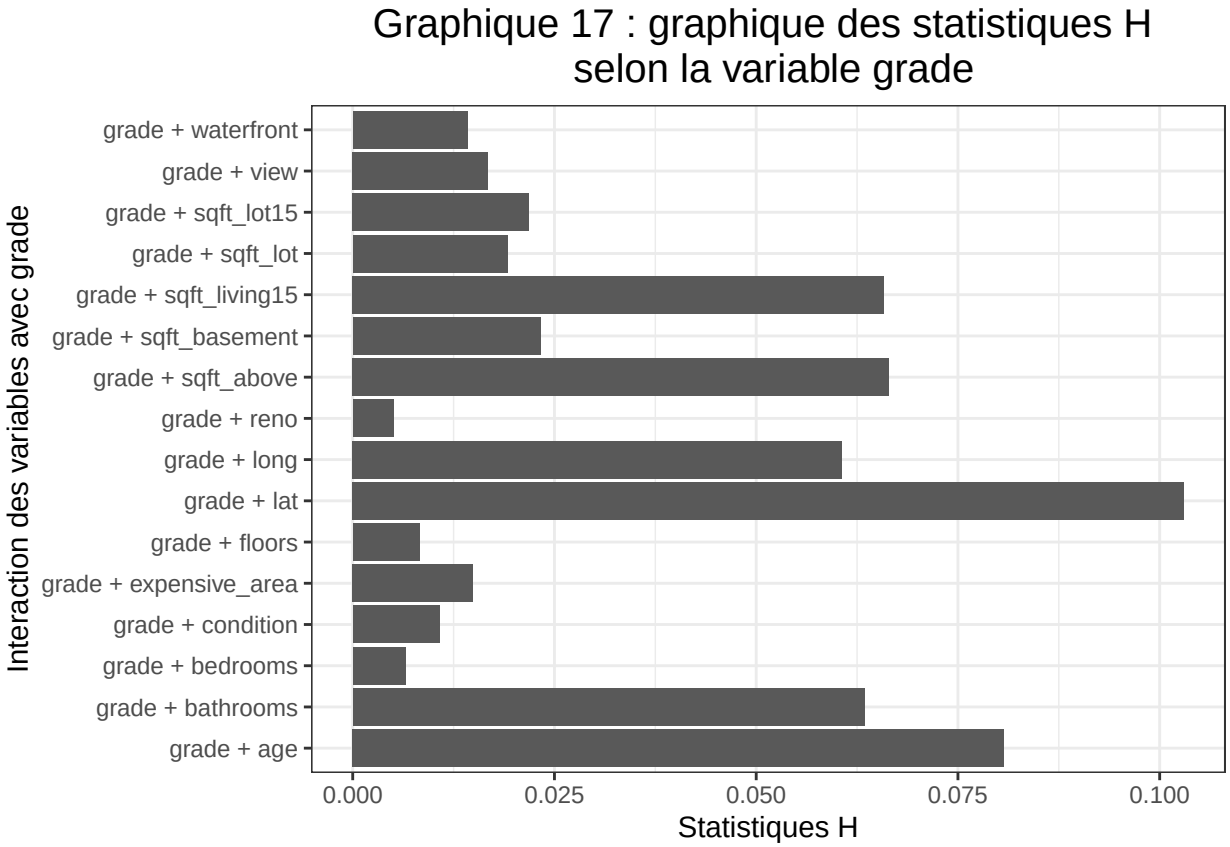
Terminons avec le PDP de la variable *sqft_above*, soit la superficie habitable au-dessus du niveau du sol des maisons.

Graphique 16 : graphique de dépendance partielle de la variable sqft_above



Cette variable n'avait pas été examinée en détail lors de l'interprétation du modèle de forêt aléatoire. On observe que le logarithme moyen du prix prédit est proportionnel à *sqft_above* : plus la superficie habitable de la maison augmente, plus son prix prédit augmentera, toutes choses étant égales par ailleurs. Cette relation est logique, puisque les maisons de grande superficie sont intuitivement plus dispendieuses que les petites maisons. On voit cependant une baisse marquée du logarithme du prix de la maison autour de 5000 pieds carrés de superficie qui semble contredire la tendance observée : cette baisse est probablement causée par un surajustement du modèle aux données d'entraînement. Nous croyons que si plus de données étaient disponibles ou que le problème de surajustement était réglé, la relation observée serait monotone.

Cependant, les PDP ne permettent pas de visualiser les éventuelles interactions entre les variables. Une manière de quantifier ces interactions est de déterminer les statistiques H de Friedman : plus ces dernières sont grandes, plus l'interaction est forte. Nous allons examiner les statistiques H entre la variable *grade*, soit la variable la plus importante du modèle, et toutes les autres variables :

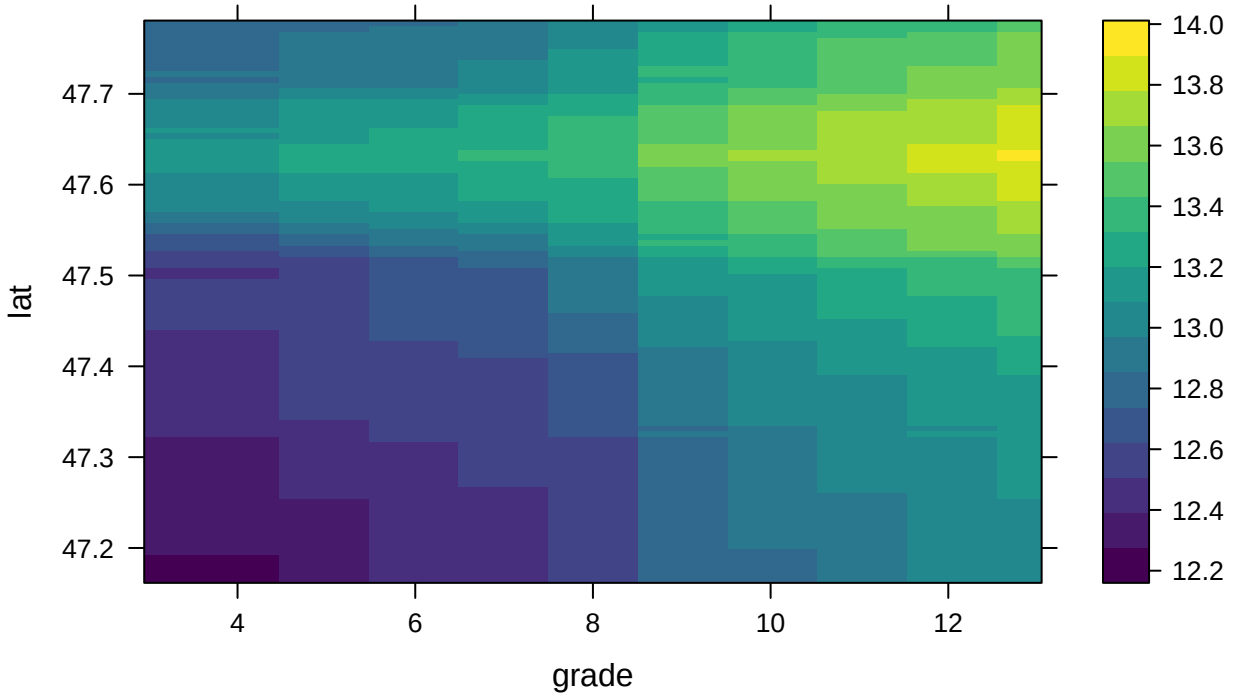


Nous remarquons qu'une seule interaction semble très forte (statistique H plus élevée que 0,1). Il y a également quelques interactions moyennes (statistique H entre 0,05 et 0,1), et nous allons considérer les autres interactions comme assez faibles.

Les deux plus fortes interactions observées sont celles entre la variable *grade* et les variables *lat* et *age*. Nous pouvons donc tracer les PDP bivariés de ces combinaisons de variables afin d'examiner plus en détail ces interactions.

Commençons par le PDP bivarié des variables *grade* et *lat*.

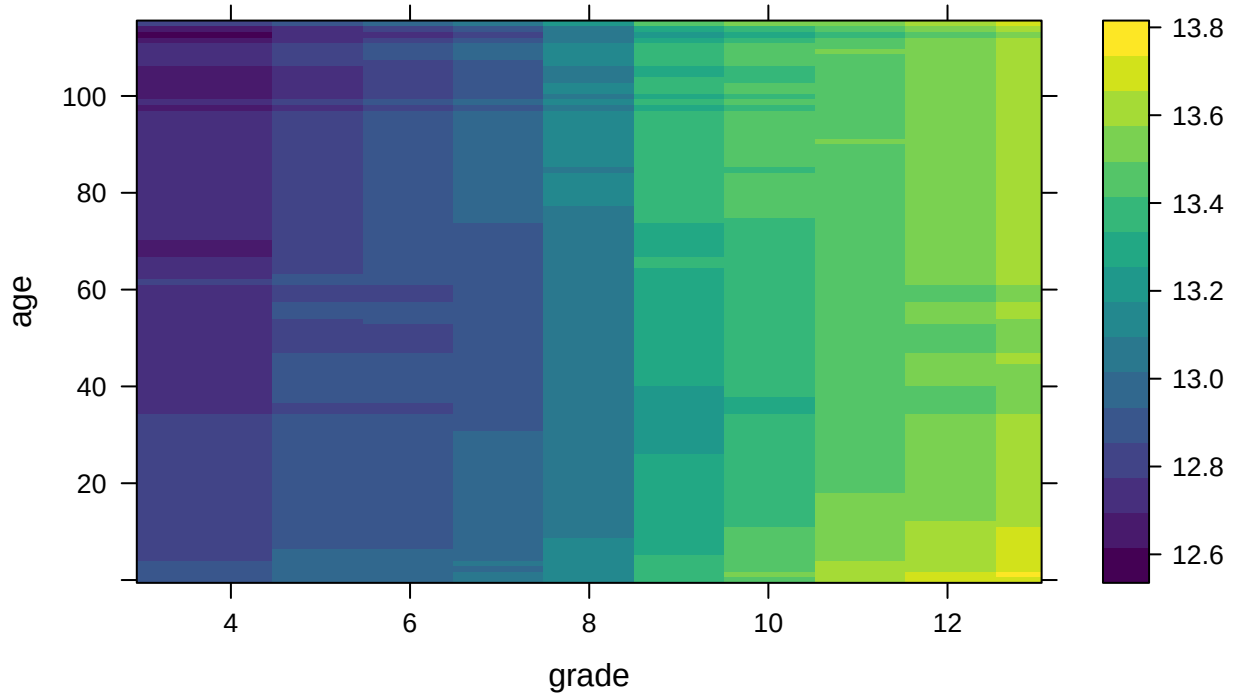
Graphique 18 : Graphique de dépendance partielle bivarié entre les variables grade et lat



Il semble en effet y avoir une certaine interaction entre ces variables : la variation du logarithme du prix prédit avec la latitude semble plus prononcée pour les mauvaises qualités de construction que pour les bonnes. Cet effet semble logique, puisqu'une maison très mal construite pourrait être extrêmement plus chère si elle est située en plein centre-ville que si elle est au milieu de la campagne, en raison de sa position prisée et de la valeur de son terrain. Des gens fortunés seraient prêts à payer très cher pour acquérir cette maison de mauvaise qualité et en construire une nouvelle sur le même terrain. En comparaison, une maison extrêmement bien construite vaudra aussi beaucoup plus cher si elle est bien située, mais intuitivement, cet écart serait moindre que pour une maison de basse qualité. On constate également avec le PDP bivarié que les maisons aux prix prédits les plus élevés sont celles de bonne qualité de construction et située à la latitude du centre-ville de Seattle. En effet, ces maisons peuvent atteindre un prix prédit de pas moins de 1,2 M $, ce qui est logique.

Enchaînons avec le PDP bivarié des variables *grade* et *age*.

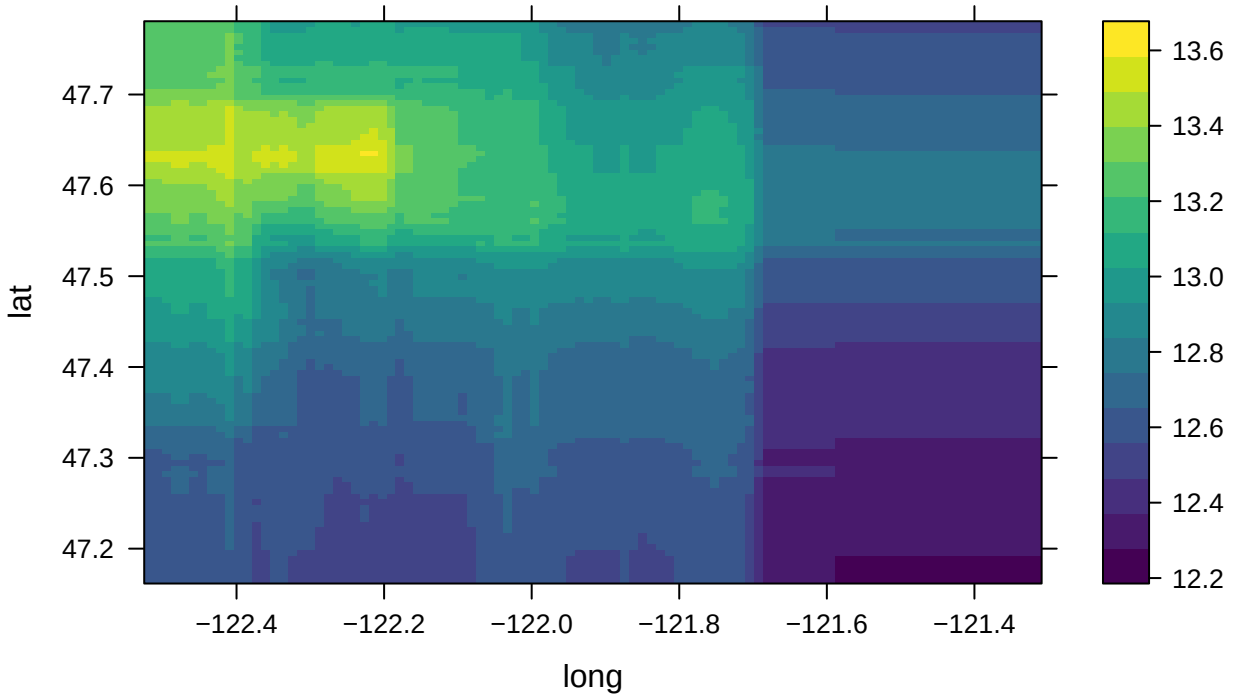
Graphique 19 : Graphique de dépendance partielle bivarié entre les variables grade et age



L'interaction entre ces deux variables n'est pas extrêmement prononcée. Nous pouvons cependant constater que pour une qualité de construction donnée, la variation du logarithme du prix prédit avec l'âge de la maison ne suit pas exactement la même tendance que pour une autre qualité de construction. En particulier, le prix prédit des maisons de qualité de construction moyenne semble varier moins intensément avec l'âge que pour les maisons de qualité mauvaise ou bonne. On constate également avec le PDP bivarié que les maisons aux prix prédits les plus élevés sont celles de bonne qualité de construction et peu âgée, ce qui semble logique.

Enfin, par curiosité, examinons le PDP bivarié des variables *lat* et *long*. Nous savons que les données spatiales ont eu une forte incidence prédictive dans notre modèle. De plus, nous avons déjà concentré une partie de notre travail à examiner cet effet prédictif (création d'une *heatmap* dans l'analyse exploratoire et dans l'interprétation du modèle de forêt aléatoire), mais nous ne l'avons pas encore fait pour notre modèle le plus puissant, le *GBM*. Pour ces deux raisons, nous croyons donc qu'il est intéressant de jeter un oeil à ce PDP bivarié pour examiner plus en détail l'effet des données spatiales dans notre *GBM*.

Graphique 20 : Graphique de dépendance partielle bivarié entre les variables long et lat



Nous observons de nouveau une *heatmap* du logarithme des prix des maisons, mais elle est mieux définie que celle obtenue avec le modèle de forêt aléatoire. En effet, les variations de prix prédit sont très graduelles et lisses par rapport à cette dernière. Nous pouvons voir très facilement la zone urbaine de Seattle et ses prix plus élevés en haut à gauche du PDP. Nous pouvons également constater une forte interaction entre les variables *lat* et *long* : pour les latitudes faibles, le logarithme du prix prédit ne varie pas beaucoup avec la longitude, alors qu'il varie beaucoup avec la longitude pour les latitudes élevées. La statistique H entre ces variables *lat* et *long* est en effet de 0,234, ce qui dénote une forte interaction entre les deux variables et confirme ce constat.

Au final, nous avons vu que le modèle de boosting de gradient stochastique est, de loin, le meilleur modèle prédictif dans notre cas. Son EQM sur les données de test est inférieure à celles de tous les autres modèles, ce qui témoigne de son efficacité. Le modèle possède donc l'avantage d'être celui qui possède le plus grand pouvoir prédictif pour notre problème.

Ce modèle n'est pourtant pas sans désavantage :

- Son temps de calcul est très long : l'optimisation du modèle a été difficile en raison des ressources de calcul disponibles limitées. De plus, seuls les deux hyperparamètres du modèle jugés les plus importants ont été optimisés, et ce pour la même raison.
- Il est difficile d'interprétation : il est difficile d'utiliser des procédés statistiques classiques pour interpréter le modèle, et des méthodes de calcul intensif doivent être utilisées.
- Nous n'avons pas obtenu d'intervalles de confiance pour les prédictions avec le présent modèle, ce qui est un fort désavantage dans une situation de gestion des risques.

6 Conclusion

Pour conclure, le meilleur modèle pour résoudre le problème est **le modèle de boosting de gradient stochastique**. En effet, il s'agit du modèle possédant la meilleure puissance prédictive parmi les sept modèles testés. Des améliorations à ce modèle sont cependant toujours possibles : si de plus grandes ressources de calcul étaient disponibles, un plus petit paramètre de régularisation λ pourrait être choisi, ce qui permettrait d'obtenir un modèle encore plus performant. Il serait également possible d'optimiser tous les hyperparamètres du modèle, ce qui résulterait sans doute en un meilleur modèle final.

Nous sommes cependant d'avis qu'avec les sept modèles étudiés, il a été possible de résoudre la problématique actuarielle, c'est-à-dire de développer un modèle performant pouvant prédire la valeur des maisons dans la région de Seattle. Ainsi, ce modèle pourra aider les assureurs de la région à avoir un meilleur portrait de la situation afin de mieux gérer leurs risques. Cependant, une des limitations du modèle est qu'il prédit le prix des maisons sur l'échelle logarithmique. Ce choix d'échelle avait pour but de faciliter la modélisation, mais pose problème lors de l'interprétation. Une piste d'amélioration serait donc de trouver une façon de ramener les prédictions à l'échelle monétaire. Pour ce faire, il faudrait estimer un paramètre de volatilité. Cette estimation, pour le modèle de boosting de gradient stochastique, dépasse les cadres du cours et n'a ainsi pas été tentée. Une seconde limitation est que notre modèle prédit la valeur des maisons sans tenir compte de facteurs importants pouvant affecter le prix des maisons dans le temps, dont l'inflation. Il serait donc intéressant de pouvoir tenir compte de l'inflation dans notre modèle afin de pouvoir l'utiliser sur des données datant de quelques années.

En somme, cette modélisation a permis de dresser un très bon portrait du marché immobilier de King County et de l'effet des différentes caractéristiques d'une maison sur son prix de vente. Elle a, entre autres, permis, à l'aide d'une carte thermique, de bien cibler les endroits plus dispendieux. Il serait intéressant d'utiliser notre modèle sur une autre région du monde, par exemple le Québec, afin de voir si le modèle s'adapte bien aux caractéristiques d'autres régions.

7 Bibliographie

1. Kaggle, harlfoxen (2017). House sales in King County, USA. Récupéré le 27 février 2020 de https://www.kaggle.com/harlfoxem/housesalesprediction.

2. Max Kuhn (2020). caret : Classification and Regression Training. R package version 6.0-85. https://CRAN.R-project.org/package=caret

3. Terry Therneau and Beth Atkinson (2019). rpart : Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart

4. Stephen Milborrow (2019). rpart.plot : Plot 'rpart' Models : An Enhanced Version of 'plot.rpart'. R package version 3.0.8. https://CRAN.R-project.org/package=rpart.plot

5. A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22.

6. Brandon Greenwell, Bradley Boehmke, Jay Cunningham and GBM Developers (2019). gbm : Generalized Boosted Regression Models. R package version 2.1.5. https://CRAN.R-project.org/package=gbm

7. Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. URL http://www.jstatsoft.org/v33/i01/.

8. Alina Beygelzimer, Sham Kakadet, John Langford, Sunil Arya, David Mount and Shengqiao Li (2019). FNN : Fast Nearest Neighbor Search Algorithms and Applications. R package version 1.1.3. https://CRAN.R-project.org/package=FNN

9. Molnar C, Bischl B, Casalicchio G (2018). iml : An R package for Interpretable Machine Learning. R package version 0.10.0. https://CRAN.R-project.org/package=iml