GLO-7027 - Analyse et Traitement des Données Massives

Partie 4 : Résultat final

Équipe 1 :

Jean-Thomas Baillargeon
Christopher Blier-Wong

Présenté le 18 avril 2018 au professeur

Richard Khoury

Département d'informatique et de génie logiciel
Faculté des sciences et de génie
Université Laval

Introduction

Ce rapport contient une discussion des modifications faites au traitement des données suite aux problématiques présentées dans la partie 3 du projet.

L'objectif est de trouver un modèle de régression qui permet de prédire le prix de vente de maisons par leurs attributs physiques. L'algorithme sélectionné prend en entrée les données d'entraînement et sélectionne le meilleur modèle d'apprentissage statistique et ses hyperparamètres grâce à une recherche en quadrille et une validation croisée à *k* plis.

Les problèmes que nous avons soulevés ainsi que les améliorations suggérées dans le dernier rapport étaient:

- Corriger le problème de convergence du SVM.
- Analyser les cas ayant les pires erreurs d'entraînement.
- Comprendre le problème de la réduction de la dimensionnalité.
- Optimiser les hyper paramètres des modèles de régression.
- Générer de nouveaux attributs spécifiques au problème.
- Lire des kernels reliés à la résolution du problème.

Le choix modèle sélectionné sera supporté par des résultats empiriques de la compétition Kaggle et les forces et faiblesses de ce modèle seront présentées.

Description de l'algorithme

L'algorithme de traitement de données sélectionné comporte trois étapes. Ces étapes sont: le prétraitement des données, la transformation non supervisée des données et le transfert supervisé de connaissances au modèle de prévision.

Prétraitement des données

L'étape de prétraitement des données (voir partie 2, les données) comprenait plusieurs éléments. Les données manquantes ont été imputées par des méthodes vues dans le cours, telle l'imputation du mode pour les données catégorielles. Ensuite, les données aberrantes pouvant nuire à l'apprentissage de notre algorithme ont été retirées. Finalement, la variable à prédire a été transformée à l'échelle logarithmique. La nouvelle distribution de la variable réponse a une forme de cloche similaire à la celle de la loi normale. Cette transformation rend les prédictions plus robustes en nivelant l'effet des données extrêmes.

Transformation non supervisée des données

Une transformation non supervisée permettant de réduire la dimensionnalité du jeu de donnée avait été envisagée lors de l'étape précédente. Plusieurs expériences ont été menées afin de valider qu'une analyse en composantes principales était bénéfique. Il a été conclu qu'il n'était pas souhaitable d'appliquer une telle transformation dans ce jeu de données. Des détails supplémentaires seront apportés dans la section résultats.

Transfert supervisé de connaissances au modèle de prévision

Afin d'obtenir une prédiction, un modèle doit être calibré. Les modèles évalués dans ce projet sont les modèles de régressions linéaires et régressions linéaires généralisées, SVM, les forêts aléatoires et *Gradient Boosting.* Le modèle *Gradient Boosting* s'est montré le plus performant et a été conservé.

Modèles linéaires

L'utilisation des modèles linéaires avait deux utilités. Premièrement l'utilisation de modèle linéaire simple permettait d'avoir rapidement une base comparative pour les autres modèles de prédiction. Il était attendu que ce modèle ne performe pas bien comparativement aux autres dues à sa simplicité. Deuxièmement, l'utilisation de modèles linéaires généralisés permettait d'obtenir un prédicteur flexible dans les situations où la distribution des variables réponses n'est pas une distribution normale. Suite à la transformation à l'échelle logarithmique de la valeur des prix de vente, la distribution est normale et un modèle linéaire généralisé n'était pas nécessaire.

SVM

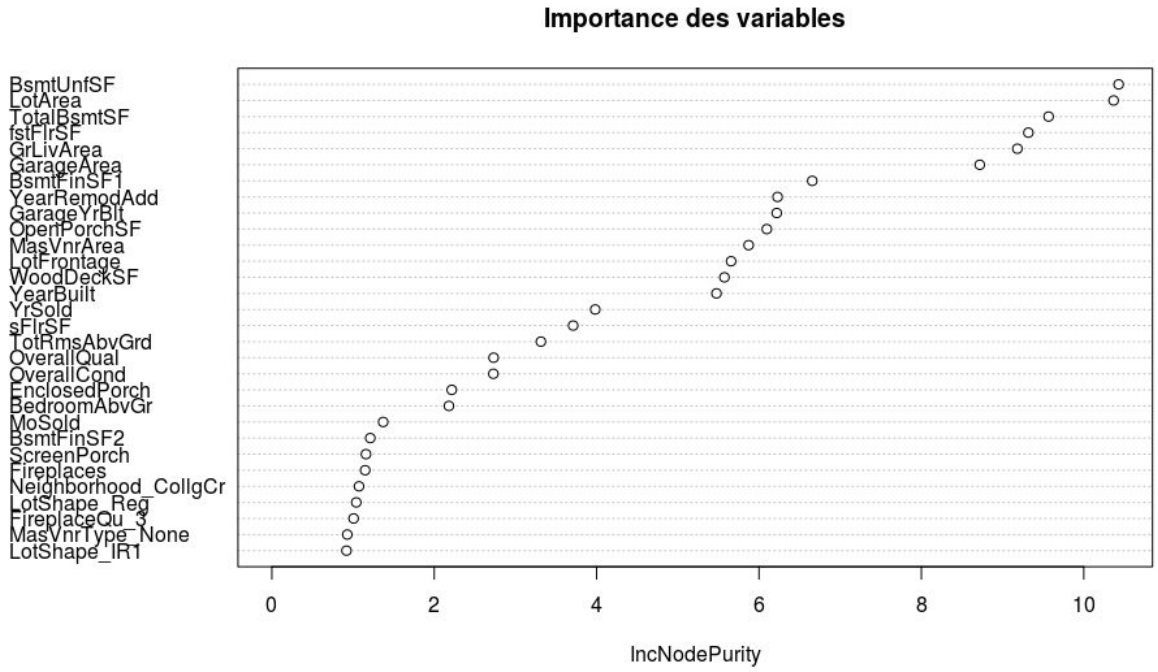
Le SVM a causé quelques problèmes en cours de projet. En effet, le modèle entraîné donnait toujours la même prédiction, peu importe les variables en entrées. Après investigation, les hyperparamètres n'étaient pas appropriés dans la recherche en quadrille. Seul un très petit ensemble de paramètres sont utiles pour faire l'apprentissage. Une fois identifiés, ils ont été incorporés dans l'algorithme sans problèmes.

Le jeu de données contenait un très grand nombre de variables explicatives pour lesquelles il était très probable qu'un SVM surapprenne. Ainsi, l'analyse par composantes principales a été utilisée afin de réduire la dimension des données. L'analyse par composantes principales produit des données linéairement indépendantes. La force du SVM est de trouver des relations non linéaires dans les données. Ainsi, comme la force du SVM ne pouvait pas s'appliquer, elle a été rejetée.

Random Forest

Le modèle de forêt aléatoire a été essayé et rejeté suite à l'analyse du graphique d'importance relative des variables. Ce graphique présente l'augmentation de l'impureté de noeuds pour les

variables utilisées dans la séparation des noeuds. Une valeur élevée dans ce graphique représente une importance élevée à cette variable dans une tâche de régression.



On remarque que les sept variables les plus importantes sont reliées à la surface de la propriété. Il s'agit d'une intuition que nous avions au début du projet. Cette grande corrélation entre les variables pose un problème lors de l'utilisation du modèle. En effet, le modèle d'arbre aléatoire est le plus efficace lorsqu'il combine plusieurs arbres indépendants entre eux. Cependant, en ayant plusieurs variables corrélées dans le jeu de données, l'effet d'ensemble (la combinaison d'arbres de régression indépendants) n'a pas pu combiner des arbres indépendants et la puissance d'agrégation des arbres n'est pas exploitée.

Gradient Boosting

Finalement, le modèle *Gradient Boosting* a été essayé et a été le plus efficace dans la tâche de prédiction des prix de vente de maison. Ce type de modèle est très performant dans plusieurs tâches et c'est pourquoi il est largement utilisé dans le cadre de compétitions Kaggle. Le principe du *Gradient Boosting* est d'appliquer un arbre de régression sur les données, et ensuite de réappliquer un arbre de régression pour corriger les erreurs des premières prédictions. Les arbres sont récursivement enchaînés jusqu'à ce que le modèle obtienne l'erreur désirée.

L'algorithme est contrôlé par deux hyperparamètres, soit le nombre d'arbres appliqués successivement et la profondeur de ces arbres. Avoir un nombre maximal d'arbres et de niveaux de profondeur est crucial afin que l'algorithme ne soit pas en situation de surapprentissage. Ces hyperparamètres ont été sélectionnés grâce à une recherche en quadrillage alimentée par des données sélectionnées par validation croisée. La valeur du

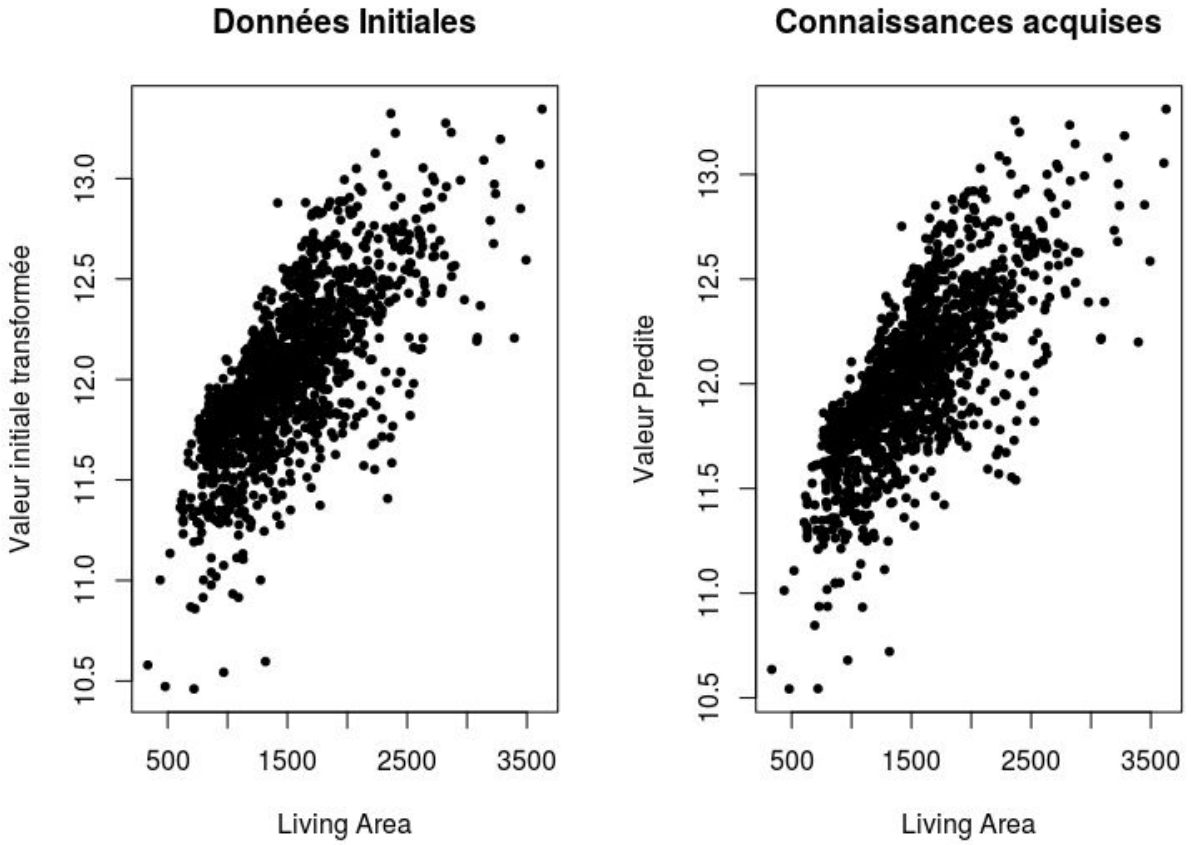
nombre maximum d’arbres est de 285 et la profondeur maximale de ces arbres est de 5 noeuds.

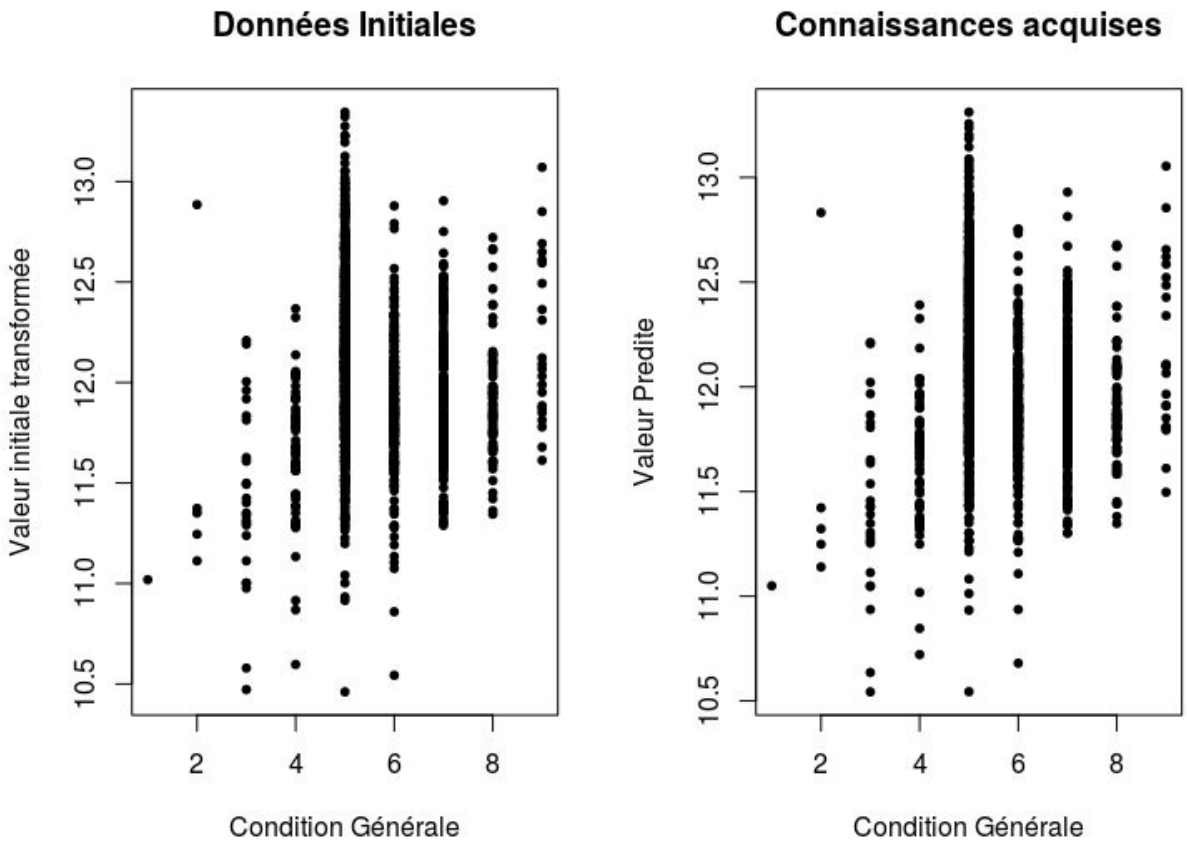
Connaissances acquises

La pertinence d’un algorithme d'apprentissage est de transformer des données en connaissances. Cet algorithme doit montrer qu’il a appris des liens cachés entre les données.

Dans le cadre de ce projet, nous avons entraîné un modèle de *Gradient Boosting* à prédire les prix de maisons selon des connaissances acquises avec le jeu de données offert dans la compétition. Initialement, lors de l’exploration des données, nous avions décelé des relations entre certaines valeurs et le prix de vente des maisons. À titre de rappel, les variables les plus corrélées étaient les superficies habitables (sous-sol, sur terre, garage…) et la condition générale de la maison.

Les deux figures suivantes présentent le lien qu’il y avait dans les données originales (à gauche) et les connaissances qui ont été acquises par l’apprenant machine (à droite).





On remarque dans ces figures la très forte similarité entre les données sur lesquelles le modèle a été entraîné et ce qu'il a prédit. Il serait possible de penser que le modèle a tout simplement surappri. Tel qu'énoncé précédemment, les hyperparamètres ont été sélectionnés en utilisant une validation croisée de telle sorte que le surapprentissage est peu probable. Autrement dit, même si le modèle s'ajuste très bien sur le jeu de données d'entraînement, les connaissances acquises généralisent bien le problème.

Présentation des résultats

Lors de la réalisation du projet, beaucoup d'expériences ont été menées afin de sélectionner les pièces optimales de l'algorithme. Les expériences ont été comparées entre elles en utilisant le score obtenu (*Root mean square error* - RMSE) lors des soumissions des prédictions sur le site de la compétition. Ces résultats auront permis d'invalider l'utilisation de l'ACP et de choisir le *Gradient Boosting* comme modèle optimal.

Invalidation de l'utilisation de l'ACP

L'utilisation de l'ACP était envisagée afin de réduire la dimensionnalité du jeu de données. Cependant, les résultats suivants ont démontré que cette réduction de dimensionnalité impactait négativement de façon significative les résultats à la compétition.

Le tableau suivant présente les résultats pour la compétition pour 3 modèles avec et sans ACP.

	RMSE	
Modèle	**Sans PCA**	**Avec ACP**
Modèles linéaires	0.13818	0.21266
Arbres Aléatoire	0.14520	0.19340
Gradient Boosting	0.12313	0.19205

Ces résultats sont a priori très surprenants. L'ACP est généralement un outil puissant et la détérioration de la puissance prédictive du modèle n'est pas attendue.

Pour mieux comprendre l'impact de la sélection du nombre de composantes conservées, une expérience a été faite. 10, 20 et 40 composantes principales ont été prises pour entraîner le modèle de *Gradient Boosting*. Le tableau ci-dessus présente les résultats de l'expérience ainsi que la proportion d'information conservée par les composantes principales.

Nombre composantes	**RMSE**	**Proportion information**
10	0.19205	0.9998
20	0.17009	0.9999
40	0.14164	0.9999
Toutes	0.12313	1.0000

Ces résultats sont étonnants et permettent de conclure qu'il y a beaucoup d'information non linéaire dans les données. Malgré que certaines variables soient peu informatives, elles sont essentielles à la modélisation du prix de vente des maisons. En omettant ces informations, le modèle se trouve dans une zone de sous apprentissage.

Sélection du modèle

La sélection préliminaire du modèle de prédiction a été faite avec les résultats suivants.

Modèle	**RMSE**
Modèles linéaires simples	0.13818
Modèles linéaires généralisés	0.43026
Arbres Aléatoire	0.14520
SVM	0.18294
Gradient Boosting	0.12313

La suite du projet a été faite avec le modèle de *Gradient Boosting* car il a obtenu le score le plus bas pour la compétition.

Afin d'améliorer le modèle, des idées venant de différents kernels de discussions ont été essayées. Un Kernel [House prices: Lasso, XGBoost, and a detailed EDA] proposait de faire une moyenne de différents modèles. Un autre kernel [Stacked Regressions to predict House Prices] proposait d'utiliser une fonction de perte différente pour le *Gradient Boosting*, soit la fonction "huber" permettant de réduire l'impact des données extrêmes. Les résultats des deux suggestions sont présentés dans le tableau ci-dessous.

Modèle	**RMSE**
Moyenne des modèles	0.12408
Fonction de perte Huber	0.12197

L'utilisation de la fonction de perte Huber a été conservée dans le modèle final.

Le modèle a été considéré satisfaisant à ce point, car les autres améliorations suggérées par les kernels étaient de l'ajustement extrême pour lesquels les modèles perdaient de leur interprétabilité et de leur valeur pédagogique.

Évaluation du modèle

Afin d'évaluer la qualité de notre modèle, Il est possible de l'évaluer autant sur une base quantitative grâce à des statistiques qu'à l'aide d'analyse de cas.

Évaluation quantitative par statistiques

La qualité du modèle de prédiction peut être quantifiée par des statistiques descriptives. En effet, la moyenne des erreurs sur le jeu d'entraînement est de -250$ par maison (ce qui est très

près de 0) et a un écart-type d'environ 3750$. Les prédictions du modèle seront à l'intérieur d'une fourchette de ±22 000$, et ce 95% du temps. En considérant que le prix moyen des maisons est de 180 000$, l'erreur moyenne de prédiction est 0% et prend une valeur au plus de ±12% de la maison, 95% du temps.

Évaluation qualitative par analyse de cas

L'analyse de cas pour le modèle sélectionné est très peu intéressant. En effet, le modèle sélectionné corrige les erreurs au fur et à mesure jusqu'à ce qu'il se rende à la limite du surapprentissage. Le modèle est donc bon à peu près partout. Si on tentait de trouver des cas où il ne fonctionne pas, cela reviendrait à exécuter l'algorithme interne du modèle une itération supplémentaire et à se rendre dans la zone de sur-apprentissage.

Pour des fins pédagogiques, une expérience a été faite afin de valider que l'information restante dans les données était très peu intéressante. La première étape de l'expérience a été de corréler les erreurs de prédiction avec différentes variables du jeu de données. Le tableau ci-dessus présente les 5 corrélations positives et négatives les plus importantes.

Plus grandes corrélations négatives

YearRemodAdd	-0.3774201053
OverallQual	-0.3647971306
YearBuilt	-0.3099196205
CentralAir_1	-0.2926501443
Foundation_PConc	-0.2765454195

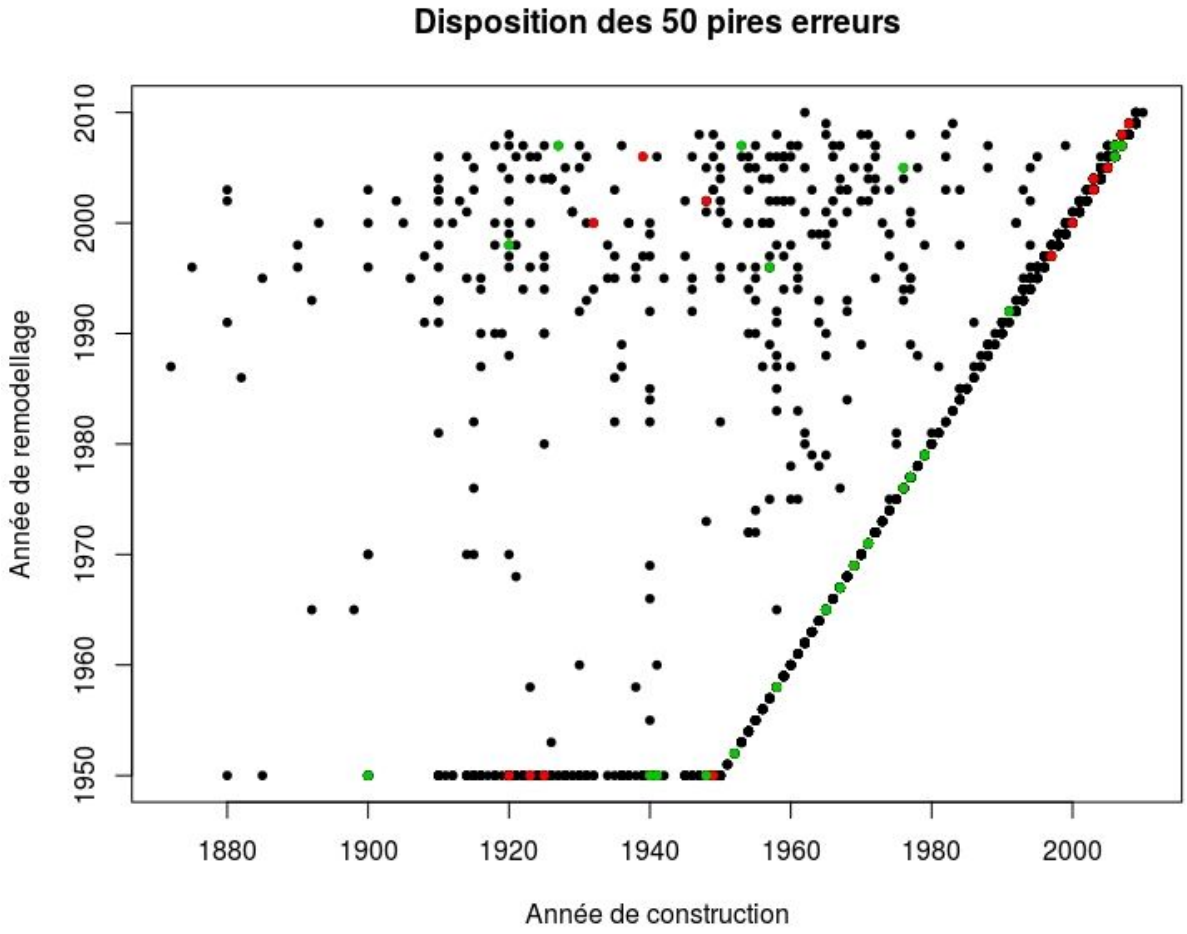
Plus grandes corrélations positives

KitchenAbvGr	0.241764213
GarageFinish_1	0.2560303732
ExterQual_3	0.2676202497
KitchenQual_3	0.2799252336
CentralAir_0	0.2926501443

On remarque que les variables ayant le plus de corrélations sont des éléments très précis d'une maison. Il est questionnable d'affirmer que le modèle sous-estime le prix des maisons n'ayant pas de système de climatisation centrale. L'expérience a été menée jusqu'à la fin et les

maisons associées aux 10 prédictions les plus sous-estimées ont été analysées. Surprenamment, aucune de ces 10 n'avait de système de climatisation centrale.

On remarque aussi que YearBuilt et YearRemodAdd, soit l'année de construction et l'année de remodelage, impactent inversement l'erreur. Ainsi le modèle prédira un prix trop haut à une maison plus ancienne et vice versa. Le graphique ci-dessous présente ces deux valeurs pour toutes les maisons et met en surbrillance les 25 maisons les plus surévaluées (en vert) et sous évaluées (en rouge.)



Ce graphique ne semble pas présenter de patron pour les sur et sous estimations du prix de vente des maisons.

En conclusion à cette évaluation du modèle, il n'est pas réellement possible de donner des endroits où le modèle fonctionne mieux et moins bien. Les pires erreurs ont été corrigées 285 fois et les relations pouvant être trouvées se trouvent à la marge du sur-apprentissage. Identifier ces relations revient à trouver une zone inefficace du modèle qui ne se généralise pas sur des données n'ayant jamais été observées.

Christopher Blier-Wong / Jean-Thomas Baillargeon

Rétrospective

Ce projet a été très intéressant sur le plan pédagogique. Malgré que certains points ont confirmé des éléments que nous connaissions déjà, nous avons été mis à l'épreuve pas d'autres éléments

Par exemple, nous avions l'intuition que le modèle *Gradient Boosting* offrait une excellente performance. Notre intuition était correcte lorsque nous pensions que la superficie habitable de la maison serait un élément clé dans le prix de vente.

Nous avons été cependant très surpris d'apprendre que les SVM ne performent pas très bien en contexte d'utilisation d'une analyse par composantes principales. De plus nous avons été très surpris que cette analyse ne soit pas bénéfique à tout coup.

Si le projet était à refaire, nous aurions probablement refait de façon similaire. Les obstacles que nous avons rencontrés nous ont permis de nous questionner sur les bases des modèles d'apprentissage automatique et d'explorer leurs limites. Il s'agit d'un apprentissage très pertinent.

Conclusion

Au final, ce projet nous a aidés à comprendre l'utilisation de modèles d'apprentissage statistique dans le contexte de la régression. Nous avons bâti un canal permettant de sélectionner le modèle le plus performant par validation croisée et recherche d'hyper paramètres optimaux par quadrillage, ce qui est une expérience importante pour un scientifique des données.

Notre modèle nous permet d'avoir un score de 0.12197, qui nous place à la position 999 sur 5087. Malgré que ce score n'est pas dans la fourchette initialement souhaitée, nous sommes très satisfaits de l'ajustement du modèle. Le code informatique est disponible en suivant ce lien GitHub [https://github.com/jtbai/glo-7027].