

Projet d’estimation des prix d'immobilier

Rapport 3 : Premier traitement des données

GLO-7027 Analyse et traitement de données massives

Ali ASSAFIRI 111 054 128

Rhita OULIZ 111 082 917

16 mars 2019

Table des matières

[**1.** **Introduction** 3](#_Toc4593293)

[**2.** **Méthodologie** 3](#_Toc4593294)

[**2.1.** **Description de l’algorithme** 3](#_Toc4593295)

[**2.2.** **Description du fonctionnement de la méthode Random Forest** 3](#_Toc4593296)

[**2.3.** **Décisions de design et d’implémentation** 4](#_Toc4593297)

[**3.** **Expérimentation** 5](#_Toc4593298)

[**3.1.** **Description des tests** 5](#_Toc4593299)

[**3.2.** **Les résultats obtenus et statistiques pertinents** 5](#_Toc4593300)

[**3.3.** **Description des résultats et discutions** 6](#_Toc4593301)

[**4.** **Discutions** 8](#_Toc4593302)

[**4.1.** **Attributs utilisés** 8](#_Toc4593303)

[**4.2.** **Attributs importants** 8](#_Toc4593304)

[**4.3.** **Comparaison avec les attributs importants du rapport 2** 9](#_Toc4593305)

[**5.** **Conclusion** 10](#_Toc4593306)

[**5.1.** **Problèmes identifiés** 10](#_Toc4593307)

[**5.2.** **Solutions** 10](#_Toc4593308)

[**6.** **Annexe** 11](#_Toc4593309)

Table des tableaux

[Tableau 1 - Statistique obtenue avec le jeu de données sources modifié 5](#_Toc4593529)

[Tableau 2 - Statistique obtenue avec le jeu de données sources transformé réduit 6](#_Toc4593530)

[Tableau 3 - Les variables retirées des données sources 6](#_Toc4593531)

[Tableau 4 - Le prédicteur performant P2 et le mauvais prédicteur P1 6](#_Toc4593532)

[Tableau 5 - Statistiques des distributions des prix connus et prédits 7](#_Toc4593533)

[Tableau 6 - Les variables retirées des données sources 8](#_Toc4593534)

[Tableau 7 - les variables les plus importantes pour prédire le prix de vente d’immobilier 9](#_Toc4593535)

[Tableau 8 - les variables les plus corrélées avec le prix de vente d’immobilier. 9](#_Toc4593536)

Table des figures

[Figure 1 - Distribution des prix d'immobilier connus et ceux prédis avec P1 et P2 7](#_Toc4593524)

[Figure 2 - Exemple d'arbre de décision 11](#_Toc4593525)

[Figure 3 - Distribution des prix d'immobilier de l'ensemble d'entrainement 11](#_Toc4593526)

[Figure 4 - Distribution des prix d'immobilier estimé avec le meilleur prédicteur 12](#_Toc4593527)

[Figure 5 - Distribution des prix prédis avec un mauvais réducteur 12](#_Toc4593528)

# **Introduction**

Dans le cadre du projet de prédiction des prix d’immobilier, le présent rapport représente le premier algorithme de traitement de données, sa description et ses résultats.

Pour cette étude, nous disposons d’un jeu de données ayant environ 81 variables de toutes sortes : continues, catégoriques, nominales, etc. Ces Attributs sont en lien avec les caractéristiques possibles d’une maison dans la ville d’Âmes aux États-Unis. Le but sera d’estimer avec une précision acceptable le prix des maisons en se basant sur leurs caractéristiques. Et ce, en déterminant leurs similarités par rapport aux maisons dont le prix est connu.

Il faudra donc produire un modèle qui trouve le meilleur lien entre l’endogène et toutes les autres exogènes. Cependant, pour éviter de construire un modèle par nous-mêmes sachant que nous avons accès aux librairies externes, nous choisissons le modèle de foret aléatoire de la libraire «scikit-learn» en python.

# **Méthodologie**

### **Description de l’algorithme**

L’extraction des connaissances à partir d’une base d’informations, intitulé « Data Mining », demande non seulement l’analyse et la validation des données, mais aussi une intuition par rapport au choix d’algorithme de prédiction. Ce dernier peut être supervisé ou non supervisé. Vu que nos données sont étiquetées avec les prix des maisons à prédire, nous poursuivons avec un algorithme supervisé.

Après avoir pris le type d’algorithme en considération et en se basent sur la revue de littérature et des expériences antérieures, l’algorithme mis sous épreuve est : Les forêts aléatoires (anglais : Random Forest). Cet algorithme supervisé peut être utilisé à la fois pour la régression et la classification. Dans le cas présent, on utilise que la régression puisque nous voulons avoir une prédiction numérique en tant que résultat final.

La méthode des forets aléatoire consiste à générer aléatoirement plusieurs générations d’arbre de décision. Il s’agit d’un algorithme hiérarchique récursif descendant dont les tests au niveau des nœuds se basent sur l’entropie, le ratio de gain d’information ou l’index de Gini. La figure 2 à l’annexe illustre un exemple d’arbre de décision.

La méthode des arbres de décision est un algorithme sensible à tout élément perturbant l’apprentissage. Ainsi, la génération aléatoire de plusieurs arbres de décision pour récupérer la moyenne pondérée des résultats permet de marginaliser les erreurs dues aux mauvais apprenants.

### **Description du fonctionnement de la méthode Random Forest**

Le principe général de l’algorithme des forêts aléatoires est le suivant : « Plus il y a d'arbres dans une forêt, plus cette dernière est robuste ». Les forêts aléatoires créent des arbres de décision sur des échantillons de données sélectionnés de manière aléatoire. Par la suite, les forêts aléatoires obtiennent des prévisions de chaque arbre et choisissent la meilleure solution par la moyenne des votes. Ces votes sont évidemment influencés par la pertinence des attributs soumis. Techniquement, l’algorithme est constitué de deux grandes phases.

La première est de questionner les données pour tirer une connaissance. Avec la technique d’échantillonnage Bootstrap, des échantillons de données sont aléatoirement pris de la base de données. Pour chaque échantillon, un arbre est généré en divisant le plus possible les données pour avoir des ensembles homogènes. L’algorithme émet des arbres de décision individuels en se basant sur un indicateur de sélection d'attributs tels que le gain d'informations, le rapport de gain et l'index de Gini pour chaque attribut. C’est ainsi que les forêts aléatoires sont créées.

La deuxième phase est de tirer des conclusions de l’apprentissage fait sur les données traitées par l’algorithme en passant soit au vote sur la pertinence des classes à utiliser pour se rendre à la bonne réponse ou dans le cas d’une régression, la moyenne pondérée de toutes les sorties d'arborescence est considérée comme le résultat final.

Les composantes clés de cet algorithme sont la taille des arbres et le nombre d’arbres à générer aléatoirement ainsi que le type de test utilisé au niveau des nœuds (à titre d’exemple : Index de Gini, gain d’information ou entropie). La taille des arbres peut être limitée soit la profondeur des arbres ou le nombre de feuilles maximal.

### **Décisions de design et d’implémentation**

Vu que cet algorithme est déjà implanté par une libraire appelée « scikit-learn » en python, il suffit en faire une bonne utilisation après avoir préparé les données. Cela étant dit, une étape préliminaire est de préparer les données pour quelle soit admissible dans l’algorithme des forêts aléatoires. Sachant que, certains hyperparamètres de la régression peuvent être choisis tel que mentionné précédemment. Les deux plus importants paramètres sont le nombre d’estimateurs et le maximum de profondeur.

Quant au nombre d’estimateurs, ce paramètre s’occupe de spécifier le nombre d’arbres dans une forêt. La valeur par défaut est de 10 arbres par forêt. Ce chiffre entre autres impacte indirectement le nombre de solutions qui seront proposées par l’algorithme, ce nombre est inversement proportionnel au nombre de variables en entrée.

Comme son nom l’indique, le maximum de profondeur se résume à être un entier qui détermine la profondeur des arbres dans chaque nœud. Autrement dit, si la valeur est None, les nœuds sont développés jusqu'à ce que toutes les feuilles (sous-branches) soient pures ou que toutes les feuilles contiennent moins d'échantillons possibles ou selon un nombre prédéterminé.

C’est deux paramètres ne peuvent être choisis arbitrairement, ainsi, des tests sont faits en variant ces paramètres et en fixant les attributs en entré, puis tester la performance du modèle en observant les valeurs de scores obtenus après chaque entrainement.

# **Expérimentation**

### **Description des tests**

Au début des traitements, les données de test et d’entrainement étaient toutes regroupées pour appliquer les mêmes manipulations qu’une seule fois. Après l’ajout et la transformation des attributs, les données sont divisées à nouveau en trois nouveaux groupes.

Le premier groupe représentant les données de tests, ils sont reconnus en les triant par l’absence de la valeur à prédire (SalePrice). Le deuxième groupe représente 30% des données restantes qui seront utiles pour des fins de validation et le troisième groupe est le reste des données 70% des données d’entrainement.

Le deuxième groupe est fait pour des fins de validation, avec peu de données et beaucoup d’attributs. Il est très fréquent que les modèles supervisés surapprenne les données et performe moins bien sur les données que nous n’avons pas accès. Avoir garder de côté des données qu’on connait leurs valeurs à prédire, nous permettra de réévaluer notre performance vue que le taux d’apprentissage sur les données d’entrainement risque d’être élevé.

Puisque nous avons qu’un seul algorithme à tester, nous avons décidé de faire plusieurs tests en variant les hyperparamètres de l’algorithme. Cette tache dépend grandement de la performance machine en fonction des données sous étude et les hyperparamètre. Il suffit alors d’observer les résultats pour en sortir avec une compréhension des résultats après avoir compris la partie technique de l’algorithme.

En revanche, des questions sont relevées à la suite d’une analyse première des résultats obtenus. Nous décidons de faire varier les données en supprimant tout court les données ayant des valeurs manquantes, de plus, ayant en même temps une corrélation négative avec l’attribut du prix de la maison. Donc, on fait varier les deux hyperparamètres simultanément vers une hausse graduelle et nous observons le score sur les mêmes données.

### **Les résultats obtenus et statistiques pertinents**

Pour les tests sur les données enrichies par le remplacement des valeurs manquantes, une amélioration est remarquée du taux de précision entre l’expérience 1 et l’expérience 2. Cependant, entre l’expérience 2 et l’expérience 3 l’augmentation des taux de précision est légère.

|  |  |  |  |
| --- | --- | --- | --- |
| Statistiques d’apprentissage | Expérience 1 | Expérience 2 | Expérience 3 |
| Profondeur maximale | None | 10 | 20 |
| Nombre d'arbres de décision par forêt | 10 | 100 | 1000 |
| Taux de précision au test | 97,58% | 97,48% | 97,80% |
| Taux de précision à la validation | 84,47% | 86,23% | 86,25% |
| Taux de précision sur toutes les données | 90,81% | 93,73% | 94,15% |
| Ratio de succès de Kaggle | **0,3286** | **0,1611** | **0,1498** |

Tableau 1 - Statistique obtenue avec le jeu de données sources modifié

En ce qui concerne la deuxièmement partie d’expérimentation, les tests sont faits sur la base donnée sans les attributs aillant un nombre de valeurs manquantes très élevé. Lors de cette expérimentation, les taux de précision augmentent à l’exception de du taux de validation qui a diminué l’expérience 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Statistiques d’apprentissage | Expérience 1 | Expérience 2 | Expérience 3 |
| Profondeur maximale | None | 10 | 20 |
| Nombre d'arbres de décision par forêt | 10 | 100 | 1000 |
| Taux de précision au test | 96,31% | 98,17% | 98,08% |
| Taux de précision à la validation | 84,59% | 75,44% | 87,81% |
| Taux de précision sur toutes les données | 92,71% | 90,96% | 94,92% |
| Ratio de succès de Kaggle | None | None | **0,1485** |

Tableau 2 - Statistique obtenue avec le jeu de données sources transformé réduit

Une comparaison entre les deux phases d’expérimentation nous permet de déterminer quel prétraitement de données utilisé et avec quels hyperparamètres notre algorithme performe.

Pour des forêts aléatoires de 1000 arbres avec une profondeur maximale égale à 20, on remarque le calcule sur les données avec ou sans enrichissement on presque les mêmes taux de précision. Comme le montre le tableau 3 ci-dessous.

|  |  |  |
| --- | --- | --- |
| Statistiques d’apprentissage | Données enrichies | Sans attributs à données manquantes |
| Profondeur maximale | 20 | 100 |
| Nombre d'arbres de décision par forêt | 1000 | 1000 |
| Taux de précision au test | 0,9783 | 0,9780 |
| Taux de précision à la validation | 0,8795 | 0,8721 |
| Taux de précision sur toutes les données | 0,9523 | 0,9424 |

Tableau 3 - Les variables retirées des données sources

Ainsi, l’algorithme performe sur la base de données réduite, et ce avec moins de calcul donc un temps de calcul plus court.

### **Description des résultats et discutions**

Dans cette partie, nous allons comparais les résultats du meilleur prédicteur de cette étude nommé P2 et un mauvais prédicteur nommé P1.

|  |  |  |
| --- | --- | --- |
| Hyperparamètres | Prédicteur P1 | Prédicteur P2 |
| Profondeur maximale | 10 | 20 |
| Nombre d'arbres de décision par forêt | 100 | 1000 |

Tableau 4 - Le prédicteur performant P2 et le mauvais prédicteur P1

La figure 1 représente les résultats de prédiction de ces deux prédicteurs. Comme on peut le constater, la distribution des prix prédits avec le meilleur prédicteur P2 a une médiane et un minimum très proche de ceux de l’ensemble d’entrainement. Pour les résultats du mauvais prédicteur P1, on remarque que la distribution prix prédite est très dissimilaire de la distribution des prix d’immobilier connus.

Le nuage de point de ces trois distributions est à l’annexe au figures 3, 4 et 5.

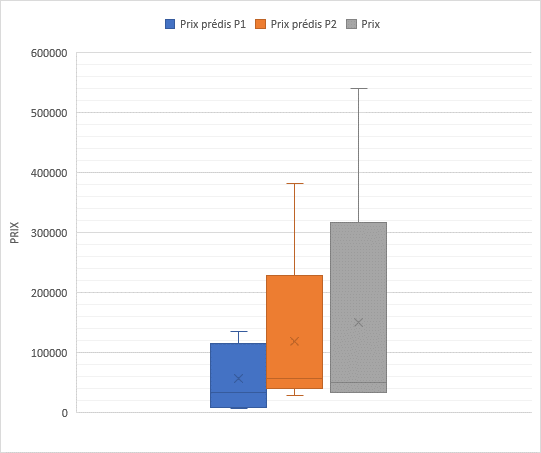


Figure 1 - Distribution des prix d'immobilier connus et ceux prédis avec P1 et P2

D’une première vue, il est clair que la distribution du prédicteur P2 est bien plus approchée de la distribution des prix connus que les distributions des résultats du mauvais prédicteur P1. Le tableau 5 ci-dessous présente les quartiles de ces trois distributions.

Les résultats du prédicteur P2 ont une trois quantiles Q1, Q2 et Q3 proches de ceux de la distribution d’entrainement.

|  |  |  |  |
| --- | --- | --- | --- |
| Quartiles | Prix prédit P1 | Prix prédit P2 | Prix connus |
| Minimum | 135751,32 | 56495,784 | 34900 |
| Premier quartile | 168703,01 | 129537,44 | 129975 |
| Médiane | 179208,67 | 157479,98 | 163000 |
| Troisième quartile | 186789,41 | 209578,40 | 214000 |
| Maximum | 281643,98 | 592694,00 | 755000 |

Tableau 5 - Statistiques des distributions des prix connus et prédits

# **Discutions**

### **Attributs utilisés**

En premier lieu, les données sont utilisées entièrement, mais des modifications étaient portées sur les données sources. Douze nouvelles variables étaient ajoutées, toutes les variables de types objets sont transformées en « dummies ». Chaque catégorie de chaque attribut nominal est rendue en soit une nouvelle colonne, autrement dit, si une colonne contient 4 types, chaque type est rendu une colonne qui contient que des zéros ou des un pour indiquer uniquement la présence de l’attribut ou son absence.

Au total, notre matrice d’entrée est de dimension : 2919 lignes X 218 colonnes. Ceci implique nécessairement que les variables les plus corrélées sont toujours les mêmes comparativement à celles obtenues dans le rapport 2 puisque ce traitement de données est fait antérieurement. Cependant, la variable la plus corrélée avec la valeur à prédire est sans doute celle qu’on a ajoutée, ce qui permet une très bonne prédiction dès le départ.

En deuxième lieu, ces données sont réduites en éliminant les attributs ayant un pourcentage de données manquantes élevé. Ainsi, la dimension de la matrice des données en entrée est de : 2919 lignes X 207colonnes, les variables suivantes étaient supprimées des données sources :

|  |  |
| --- | --- |
| Variables | Description sommaire |
| *MiscFeature* | Les ajouts non conventionnels à l’intérieur de la propriété |
| *PoolQC* | La qualité de la piscine |
| *Fence* | La qualité de la clôture |
| *FireplaceQu* | La qualité des bornes-fontaines proches |
| *Alley* | Le type des allées donnant accès à la propriété et le dernier attribut |
| *Frontage* | La route connectée à celle donnant accès à la propriété |

Tableau 6 - Les variables retirées des données sources

### **Attributs importants**

À la suite de la détermination de l’architecture interne des forêts ainsi que le nombre d’arbres dans chaque forêt, nous somme capable d’extraite le taux d’importance de chaque attribut dans le modèle à l’aide des fonctions préétablies de la librairie utilisée, ainsi, voici les 10 plus importantes variables :

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Variable | Description | Taux d’importance |
| 1 | LivingTotalSF\* | La somme de toutes les variables de superficies | 45,2% |
| 2 | OverallQual | Qualité générale du matériau et de la finition | 13,9% |
| 3 | YearBuilt | Date de construction originale | 3,6% |
| 4 | GrLivArea | Surface habitable au-dessus du niveau du sol | 2,4% |
| 5 | GarageArea | Taille du garage en pieds carrés | 1,9% |
| 6 | LotArea | Taille du terrain en pieds carrés | 1,8% |
| 7 | PercentSQtoLot\* | Le taux entre LivingTotalSF et le LotArea | 1,6% |
| 8 | YearRemodAdd | Date de remodelage | 1,6% |
| 9 | YearSinceRemodel\* | Nombre d’années depuis la date de remodelage | 1,5% |
| 10 | TotalBsmtSF | Nombre total de pieds carrés de sous-sol | 1,2% |

Tableau 7 - les variables les plus importantes pour prédire le prix de vente d’immobilier

À noter que les variables ayant un astérisque (\*) sont des variables ajoutées sur les données sources.

### **Comparaison avec les attributs importants du rapport 2**

Dans le rapport précédent, les dix variables les plus importantes avec une première analyse étaient les suivantes :

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Variable | Description | Corrélation |
| 1 | OverallQual | Qualité générale du matériau et de la finition | 79% |
| 2 | GrLivArea | Surface habitable au-dessus du niveau du sol | 71% |
| 3 | GarageCars | Taille du garage en capacité de la voiture | 64% |
| 4 | GarageArea | Taille du garage en pieds carrés | 62% |
| 5 | TotalBsmtSF | Nombre total de pieds carrés de sous-sol | 61% |
| 6 | 1stFlrSF | Premier étage pieds carrés | 61% |
| 7 | FullBath | Salles de bain complètes au-dessus du niveau | 56% |
| 8 | TotRmsAbvGrd | Nombre total de chambres au-dessus du sol | 53% |
| 9 | YearBuilt | Date de construction originale | 52% |
| 10 | YearRemodAdd | Date de remodelage | 51% |

Tableau 8 - les variables les plus corrélées avec le prix de vente d’immobilier.

On peut remarquer que les variables importantes dans le modèle des forêts aléatoires sont majoritairement présentes dans les dix variables les plus corrélées avec la variable à prédire. Toutefois, l’ordre de position n’est plus le même, certaines variables crées sont belle est bien pertinentes et utiles pour la prédiction, d’autres variables à un taux de corrélations élevé se trouvent dans le bas de l’échelle d’importance. L’hypothèse tirée pour cette partie que : la matrice de corrélation peut être utile surtout pour sa simplicité à être produite, certes, elle n’est pas suffisante pour écarter on mettre en valeur les variables qui influencent la prédiction finale.

# **Conclusion**

### **Problèmes identifiés**

En se fiant sur les résultats obtenus, on voit qu’il est possible de mieux performer si les données sont réduites à un meilleur choix concernant les variables à écarter. Il est difficile de prendre une décision éclairée à ce sujet, mais en séparent les données d’entrainement pour garder un ensemble de validation, la performance du même algorithme avec les mêmes hyperparamètres était moindre. Ceci soulève une inquiétude que le modèle surapprend très rapidement avec un petit jeu de données.

### **Solutions**

Étant donné que le jeu donné peut être sujet à amélioration, nous prévoyons pour la prochaine étape de considérer à prendre en compte la corrélation entre les variables puisqu’il y a des fonctions qui nous permet d’évaluer la pertinence de la présence des variables combinées, il existe 3160 combinaisons possibles avec les données sources, il est possible de garder que ceux qu’ils seront jugés pertinent, et ceci permet de diminuer le nombre des variables totales. De plus, au lieu de faire des « dummies » à partir des données de type objet, une matrice de similarité peut être produite pour en éliminer davantage de variables.

# **Annexe**

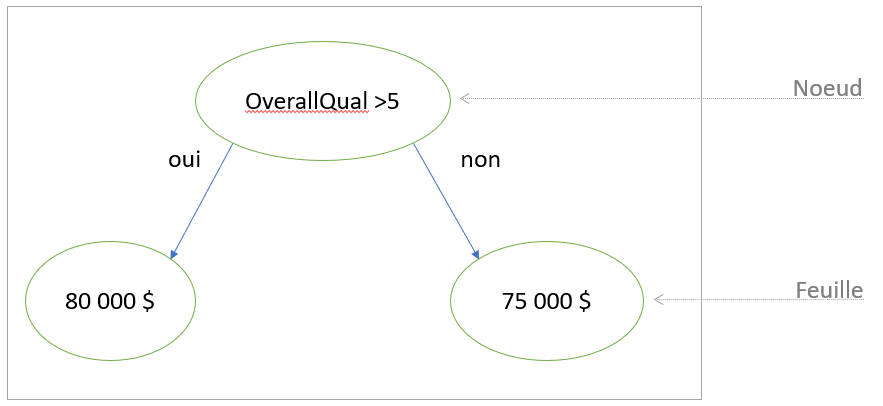


Figure 2 - Exemple d'arbre de décision

Figure 3 - Distribution des prix d'immobilier de l'ensemble d'entrainement

Figure 4 - Distribution des prix d'immobilier estimé avec le meilleur prédicteur

Figure 5 - Distribution des prix prédis avec un mauvais réducteur