RAPPORT DE PROJET

IAS

MSILINI Yassine

&

JAROSSAY Thomas

&

YILMAZ Mikail

&

TAGUENGAYTE Lina

Responsable de l'UE et chargé de projet : François Landes <rancois.landes@universite-paris-saclay.fr>

Introduction

Nous allons à travers ce projet, prédire le prix possible d'une maison dans le comté de King County à partir d'un dataset. Pour cela, nous allons tout d'abord récuperer les données pour les apercevoir, explorer les données, puis faire le preprocessing (le raffinage des données) du dataset avant d'entraîner les modèles choisis sur ces données.

Dataset

Il s'agit d'un dataset obtenu à l'aide d'une récuperation de donnée réalisée par l'open source *gis king-county open data* et est constituée de plusieurs données techniques comme le nombre de chambre et de salle etc. Ainsi que l'année de vente et sa position géographique.

Le dataset est constitué d'environ 21 263 lignes et 24 labels/colonnes avec le price au cœur de notre analyse. Les features du dataset sont en grande partie catégorielles.

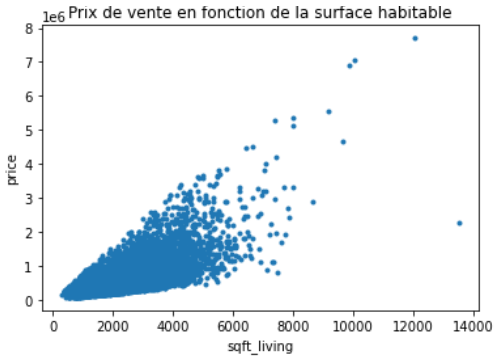
Analyse du dataset

Nous avons aperçu plusieurs choses intéressantes :

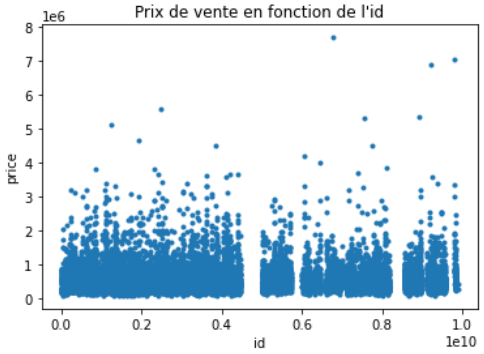
- Certaines features sont des valeurs qualitatives : waterfront, view, condition, grade et yr_renovated qui est égale à l'année de rénovation si la maison a été rénové, 0 sinon.

- Les statistiques de sqft_living, sqft_basement, sqft_lot15 semblent avoir une valeur aberrante :
  - → sqft_living: les quartiles nous indiquent que la moyenne des maisons est d'environ entre 1420 et 2550 m² d'espace habitable, avec une moyenne de 2079m² tandis que la valeur maximale est de 13540m²
  - → sqft_basement: les quartiles nous indiquent que la majorité des valeurs de cette variable est autour de 0m² et une autre autour de 560m² alors que on a une valeur maximale à 4820m².
  - → bedrooms: les quartiles nous indiquent que la majorité des valeurs de cette variable est autour de 3/4 chambres alors que nous avons une valeur maximale de 33 chambres.

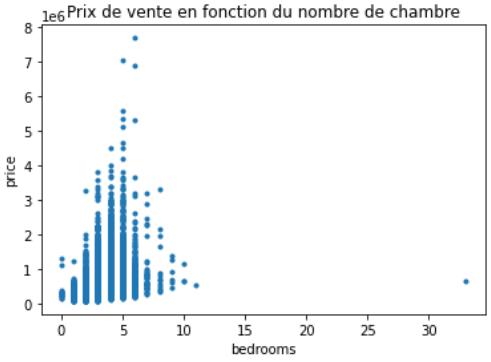
Nous avons vérifié plusieurs corrélations possibles entre les features et le price et nous avons remarqué un non corrélation avec l'id et/ou la surface habitable en m² mais une possible corrélation qui sera à affiné avec le nombre de chambre :



Remarques : - La majorité des ventes sont entre 300m² et 6000m² avec comme prix de vente entre 75000 et 280$



Nous remarquons uniquement que la majorité des ventes est en dessous de 200 000$ quelque sois l'id de vente.Il n'y a donc pas de corélation entre ces 2 variables (nous aurions pu penser que les id étaient donnée en fonction du prix de vente)



Ayant une potentielle corrélation entre le nombre de chambres et le prix, nous allons affiné la data :

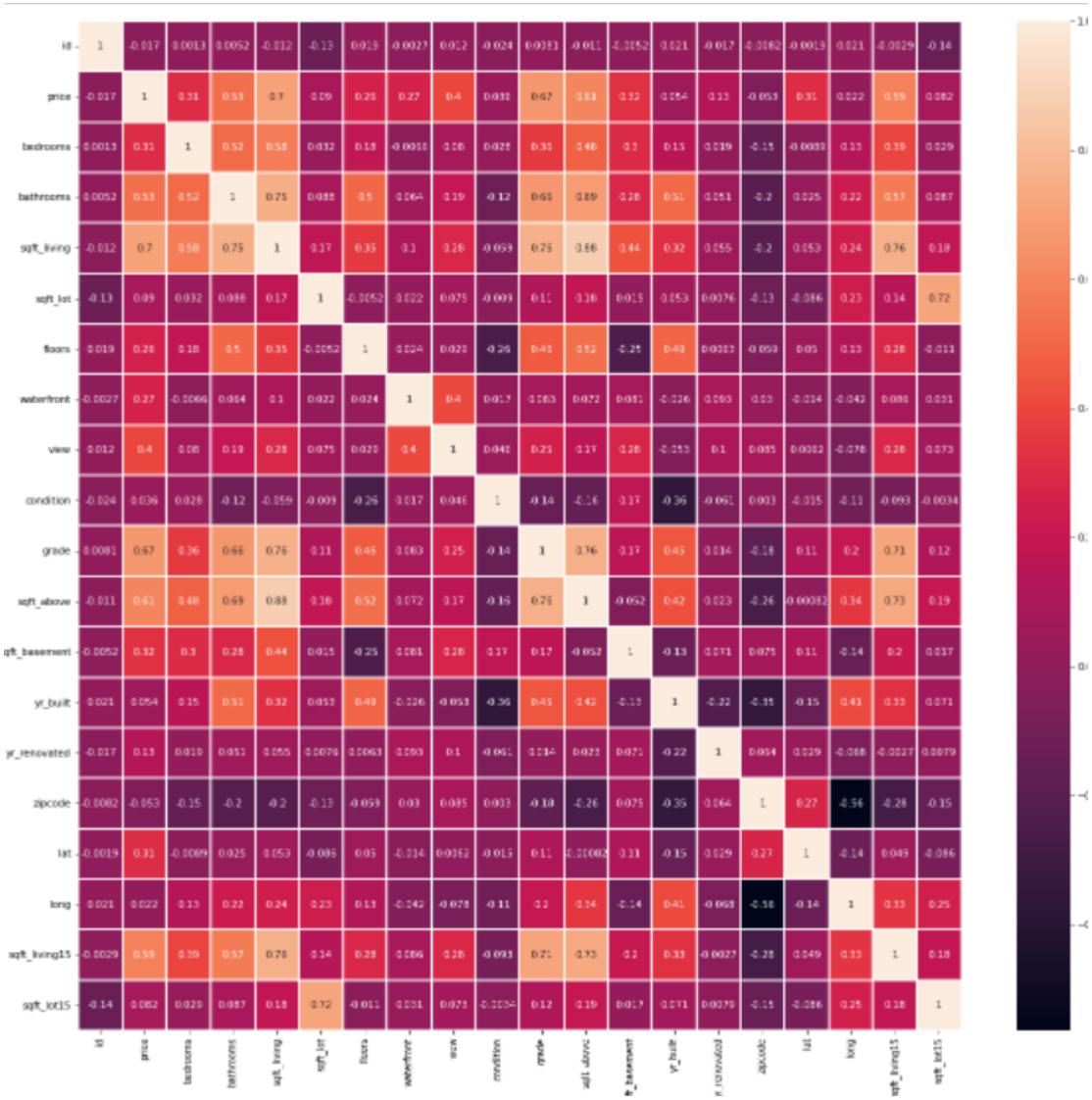
Apres avoir vu cette corrélation, un boxplot va permettre de mieux comprendre le prix, en effet, d'après celui-ci une maison atteindra un prix maximal avec 6 chambres, avec plus de chambre le prix baissera (il y a d'ailleurs une valeur aberrante qui sera supprimé qui est de 33 chambres).

Préprocessing / Raffinage du dataset

La partie preprocessing consiste à préparer aux mieux nos données afin d'optimiser les performances de l'algorithme, c'est la partie la plus importante en machine learning.

Nous avons donc, après l'exploitation de donnée, faire le raffinage de nos données avec la suppression des valeurs aberrantes, des possibles labels inutiles etc.

Nous avons dans un premier temps utilisé la heatmap qui va nous permettre de voir les différentes corrélations ainsi que les features possiblement inutiles



Nous supprimons dans un premier temps la donnée aberrante de bedrooms vu précédemment qui était de 30 bedrooms, ensuite grâce au heatmap de corrélations on voit que les features suivantes sont inutiles et elles sont donc supprimées avec la commande : .drop(columns...)

"id","sqft_lot15","long","zipcode","yr_built","condition","sqft_lot"

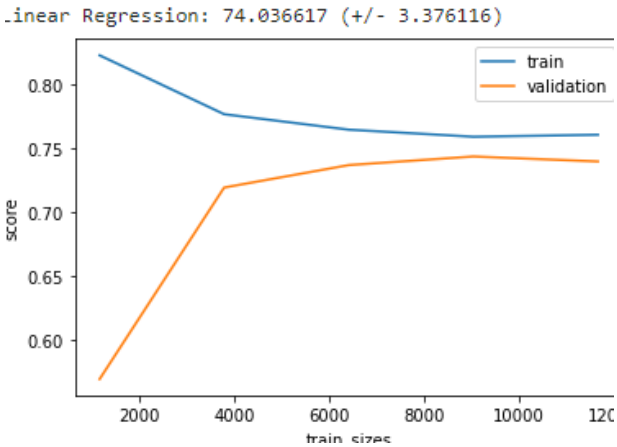
Nous devons ensuite nous occuper d'une colonne très problématique qui est la date, en effet nous allons la formater car la date se présente sous la forme de : YearMoDaT000000 qui est totalement inutilisable par un modèle, nous allons donc la mettre sous une forme de valeur entière exploitable.

Modèle

La tâche de notre modèle sera d'effectuer la régression de nos données, nous pouvons déjà éliminer tous les modèles de classification. Ensuite, vu que nous avons décidé de ne pas normaliser nos données, étape de preprocessing que certains modèles ont besoin, et que nous avons une quantité de données assez élevée (21 263). Enfin vu que nos données ont une dimensionnalité assez importante et que nous ne voulons pas perdre de précision, nous avons opté pour les modèles suivants : Linear regression, Ridge regression, Random forest regression et KNN regression.

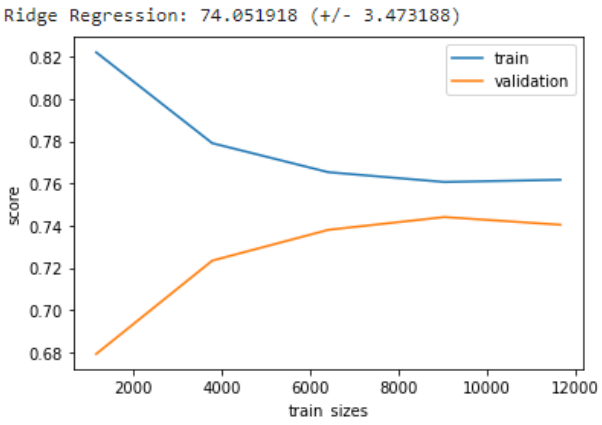
Linear régression

Le premier modèle que nous avons utilisé est le Linear regression de la bibliothèque sklearn. Dans un premier temps, ce modèle a été entraîné avec un train split de 40%, ensuite nous avons vérifié le taux d'erreur et avec le cross validation nous avons obtenu un taux de performance de 66% puis de 74% après une tentative d'optimisation via le polynôme de degré 2.



Ridge régression

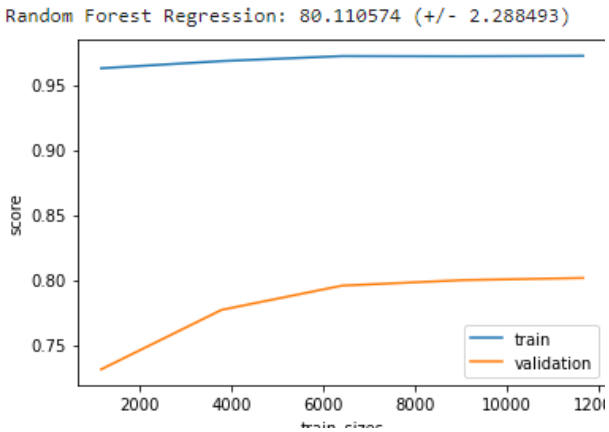
Le second modèle que nous avons utilisé est le Ridge regression de la bibliothèque sklearn. Dans un premier temps, ce modèle a été entraîné avec un train split de 40%, ensuite nous avons vérifié le taux d'erreur et avec le cross validation nous avons obtenu un taux de performance aussi de 66% puis de 74% après une tentative d'optimisation via le polynôme de degré 2.



Random Forest régression

Le troisième modèle utilisé est le Random Forest regression. Ce dernier est un modèle qui est composé de plusieurs Decision Tree, ce qui facilite l'optimisation plus tard. Nous avons donc initialisé notre modèle avec les hyperparamètres optimaux de notre modèle précédent. Ce modèle aura naturellement un meilleur score que les deux autres modèles de par son fonctionnement mais il est difficile de visualiser le procédé de régression, de plus, ce dernier aura un plus gros temps de calcul.

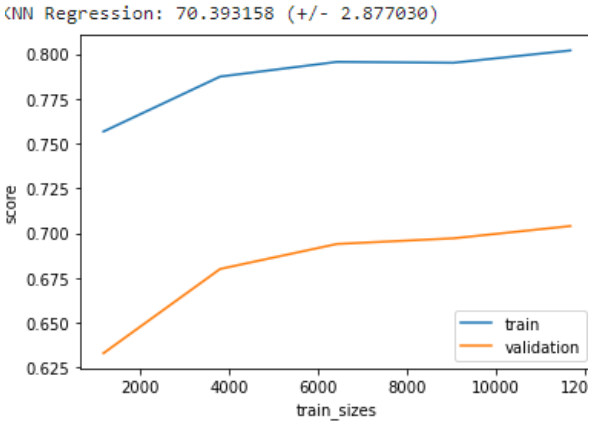
Nous avons obtenu un taux de performance aussi de 80% après une tentative d'optimisation via le polynôme de degré 2.



KNN régression

Le dernier modèle est le KNN regression qui utilise les distances vectorielles entre les différents échantillons et une constante k que l'on définit.

Nous avons obtenu un taux de performance aussi de 70% après une tentative d'optimisation via le polynôme de degré 2.



<u>Optimisation</u>

Pour finir, après comparaison des valeurs, nous avons optimisé au mieux le Random Forest en minimisant les valeurs possibles de certains labels comme : 'bootstrap', 'max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', et 'n_estimators' pour atteindre en degré 2 85%, qui, en prenant en compte le type de donnée ainsi que le nombre ligne, est un très bon score.