Lamoureux Léo 21602944
Pires Pinto Lola 21706287

ALGO : RANDOM FOREST

Etape 5

Régression :

En effectuant un premier test avec les paramètres par défaut de l'algorithme nous obtenons une précision $p \approx 75\% \leq p \leq 80\%$.

Nous allons faire varier les paramètres suivants pour tenter d'améliorer la précision du modèle :

- **n_estimators**
- **max_depth**
- **max_samples**
- **max_leaf_nodes**

Nous avons choisis n_estimators et max_depth car ils sont pour nous les deux paramètres de base de notre algorithme. De plus, lors de l'étape 3, nous avions observé des variations significatives de précision. Ce sont donc deux paramètres importants à faire varier.

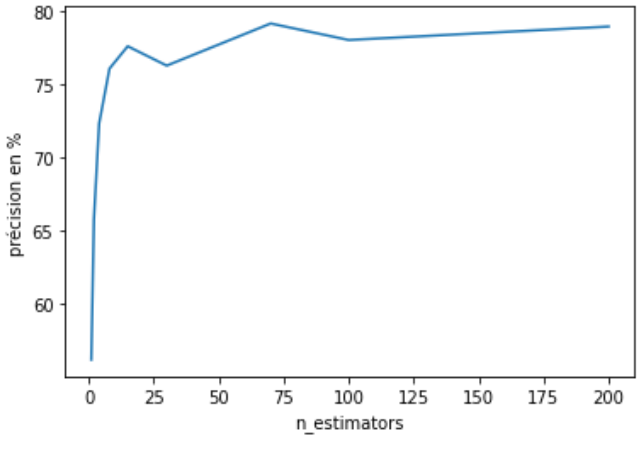
Nous avons aussi choisis max samples et max leaf nodes car nous observions aussi des variations significatives de précision en étape 3.

Nous n'utilisons pas min_sample_leaf et min_sample_split car sur nos analyse de l'étape 3, les faire varier ne faisait que diminuer la précision et la meilleur précision était obtenue pour la valeur minimale par défaut.

- **n estimators :**

Il définit le nombre d'arbres de la forêt. Par défaut il est calibré à 100, nous le ferons donc varier sur des petites valeurs, des valeurs proches de 100 et plus grande que 100.

n_estimators = [1, 2, 4, 8, 15, 30, 70, 100, 200]

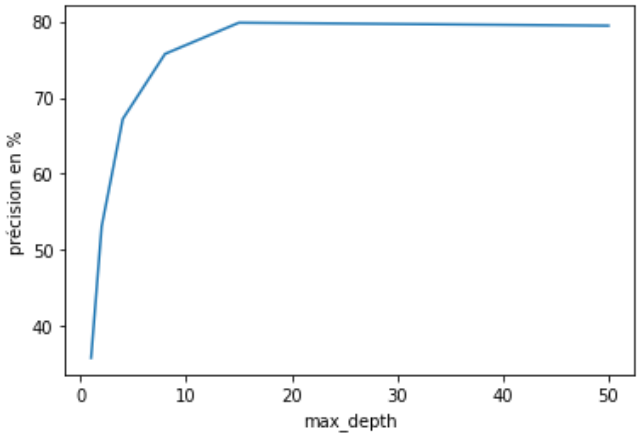


La meilleure précision semble être pour ≈ 75 arbres. La précision stagne autour de la valeur par défaut.

- **max depth :**

Il définit la profondeur maximale de nos arbres de décision. Par défaut, il se développe jusqu'à ce que les feuilles soient pures, nous voulons donc voir si un petit nombre de feuilles apporte une meilleure précision plutôt que de laisser l'arbre grandir jusqu'à ce que les feuilles soient pures.

max_depth = [1, 2, 4, 8, 15, 50]

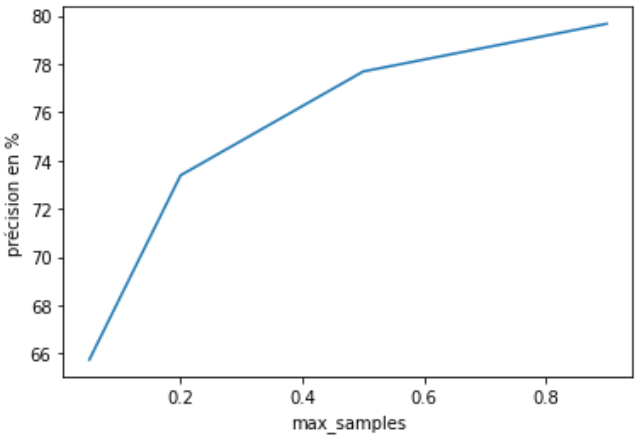


La meilleure précision semble être pour une profondeur ≥ 15.

- **max samples :**

Il définit la fraction de données du dataset qui sera utilisée par chaque arbre de décision. Nous le faisons donc varier selon des fractions de tailles du dataset.

max_samples = [0.05,0.2,0.5,0.9] (pourcentages de la taille du dataset)

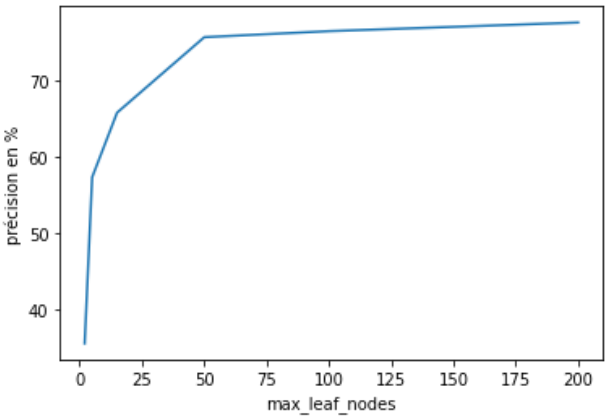


La meilleure précision semble être obtenue pour la plus grande fraction du dataset.

- **max leaf nodes :**

Il définit le nombre maximal de feuilles par arbre. Par défaut, un nombre infini de feuilles peut être développé. Nous voulons aussi voir si un petit nombre de feuilles apporte plus de précision qu'un trop grand nombre de feuilles possible.

max_leaf_nodes = [2, 5, 15, 50, 100, 200]



La meilleure précision semble être obtenue pour un grand nombre de feuilles, laisser le paramètre par défaut peut être une bonne solution, car au-dessus de 50, la précision continue d'augmenter mais très légèrement.

Nous allons donc calibrer le Grid Search selon ces intervalles :

$$50 \leq n\_estimators \leq 200\,,\ 15 \leq max\_depth \leq 30,\ 0.7 \leq max\_samples \leq 0.9,$$
$$50 \leq max\_leaf\_nodes \leq 200$$

Résultats :

***n_estimators**: 125,*
***max_depth**: 20,*
***max_samples**: 0.9000000000000001,*
***max_leaf_nodes**: 175*

***RMSE** = 276.68, moyenne des observation = 1055.69*
RMSE correspond à 26.21% des observations
***r2_score** = 79.30%*

On peut constater que la précision obtenue appartient à l'intervalle de précisions obtenu avec les paramètres par défaut $p \in [0.75, 0.80]$.

La précision est donc sensiblement la même avec ou sans calibrage des paramètres avec Grid Search Cv. Cela signifie que les valeurs par défaut de nos paramètres s'avèrent être optimales.
Cela semble normal car les meilleures précisions que nous observons sont obtenues pour les plus grandes valeurs des paramètres et que par défaut, ils sont développés sur des grandes valeurs.

Exemple : pour max_leaf_nodes, les arbres peuvent avoir un nombre de feuilles illimité, et on constate que c'est pour un grand nombre de feuilles que la précision est maximale. C'est donc normal qu'elle soit maximale avec le paramètre par défaut.

Classification :

En effectuant un premier test avec les paramètres par défaut de l'algorithme nous obtenons une précision $p \approx 100\%$.
Trouvant cela étrange, nous avons testé l'algorithme plusieurs fois pour voir si nous obtenions des précisions différentes. Mais même sur 100 tests, seulement 3 précisions étaient < 100% (98% chacunes).
Nous avons aussi effectué une validation croisée pour tester différentes découpes du dataset, mais nous obtenons toujours une précision égale à 100%.

Faire varier des paramètres ne peut donc que diminuer la précision de l'algorithme.

Nous ne savons pas vraiment comment interpréter ces résultats.
L'algorithme est peut-être en sur-ajustement sur les données car nous obtenons aussi une précision de 100% sur les données d'entraînement.
Ou alors l'algorithme s'adapte juste très bien pour ce type de dataset.
Que devons nous faire ?
Devons nous choisir un autre dataset pour faire varier les paramètres ?