

## 1. Principe de l'optimisation bayésienne :

L'optimisation bayésienne consiste à optimiser des fonctions coûteuses à évaluer en modélisant l'incertitude/variance de la fonction objective avec un modèle probabiliste. Elle a une approche probabiliste.

## 2. Définition et rôle des processus gaussiens :

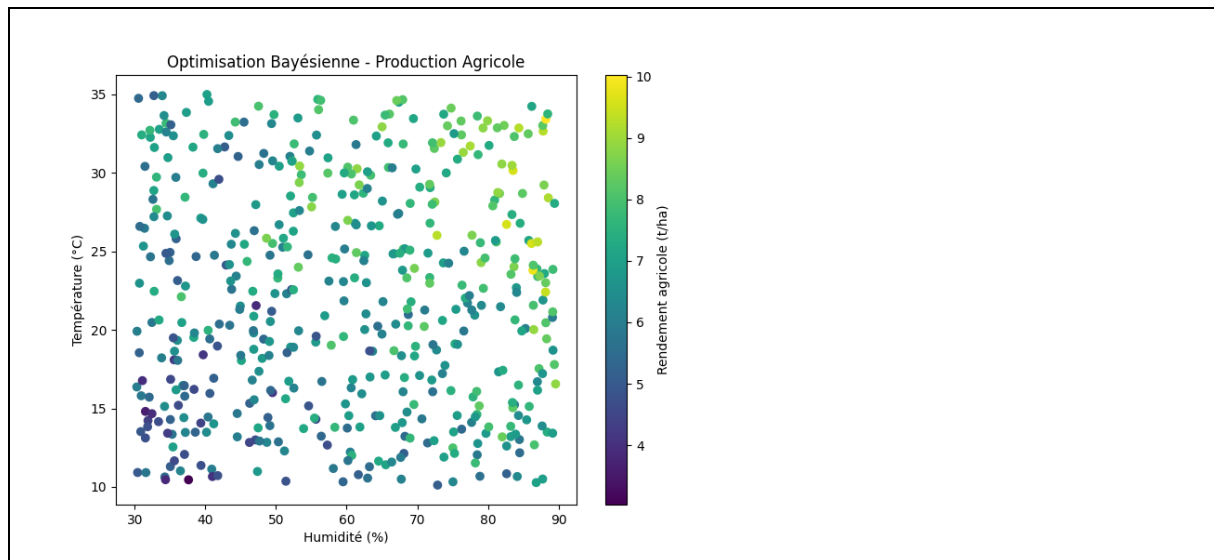
Un processus gaussien est une collection infinie de variables aléatoires avec des sous-ensemble qui suit une distribution gaussienne.

L'optimisation bayésienne, sert à modéliser l'incertitude sur la fonction objective en fournissant, pour chaque point, une estimation de la valeur moyenne et de la variance (très importante pour notre cas avec des variables qui sont incertain).

## 3. Fonctions d'acquisition principales :

- Expected Improvement (EI) : Evalue le gain potentiel en favorisant les points susceptibles d'améliorer le meilleur résultat obtenu.
- Upper Confidence Bound (UCB) : Combine la moyenne prédite et la variance avec un paramètre ajustable. Cela permet de trouver un compromis entre exploration et exploitation.
- Probability of Improvement (PI) : Mesure la probabilité qu'un point donné améliore le résultat actuel, en privilégiant les zones avec une forte chance d'amélioration.

## 4 – 5 : Code



### Interprétation :

Cette figure affiche un nuage de points qui représentent l'humidité et la température avec une coloration indiquant le rendement agricole (plus clair = plus rentable). On observe que dans le coin en bas à droite le rendement est assez mauvais pour une intervalle de 10-15 °C et 30-40% d'humidité. Et à l'opposé de la diagonale un rendement plus fort.

### Resultat Grid/random Search :

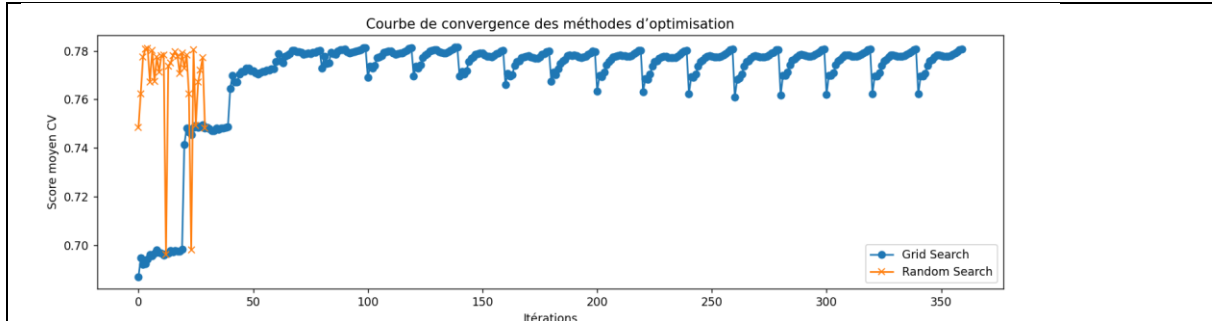
Meilleurs hyperparamètres Grid Search : {'max\_depth': 9, 'n\_estimators': 200}

Meilleurs hyperparamètres Random Search : {'n\_estimators': 190, 'max\_depth': 8}

**Interprétation :** Les deux méthodes se rapprochent : Le Grid Search recommande 200 arbres avec une profondeur de 9, et le Random Search 190 arbres avec une profondeur de 8. Cela

indique une complexité modérée, avec un nombre élevé d'arbres. C'est optimal pour notre jeu de données, on aura de bonnes performances sans surajustement.

## 6. Commentaires :



### Interpretation :

Nous avons ici une courbe de convergence, qui compare l'évolution des scores moyens obtenus lors de l'optimisation des hyperparamètres avec *Grid Search* et *Random Search*. Pourquoi nous avons ce graph ? Est bien *Grid Search* explore systématiquement toutes les combinaisons d'hyperparamètres, ce qui nous donne une courbe de convergence plutôt régulière (bleu) mais avec plus d'itération. En revanche, *Random Search* (orange), échantillonne aléatoirement et test sur un nombre plus faible d'itération un plus large panel de d'hyperparametre. C'est pourquoi nous avons plus de variation sur une itération plus faible pour le *GridSearch*. En conclusion meme si les deux convergent vers des performances similaires le random search reste tout de meme plus efficace dans notre cas. On utilisera *GridSearch* si le nombre d'hyperparametre est élever.

## 7. Avantages et limites de l'optimisation bayésienne

### Avantages :

- Très efficace pour optimiser des fonctions coûteuses à évaluer.
- Utilise l'incertitude pour guider le choix des points a tester.
- A généralement une meilleure convergence par rapport aux approches classiques (Grid/Random Search).

### Limites :

- Complexité croissante avec le nombre de dimensions.
- Sensible à la qualité des points initiaux et à la spécification du modèle probabiliste utilisé.

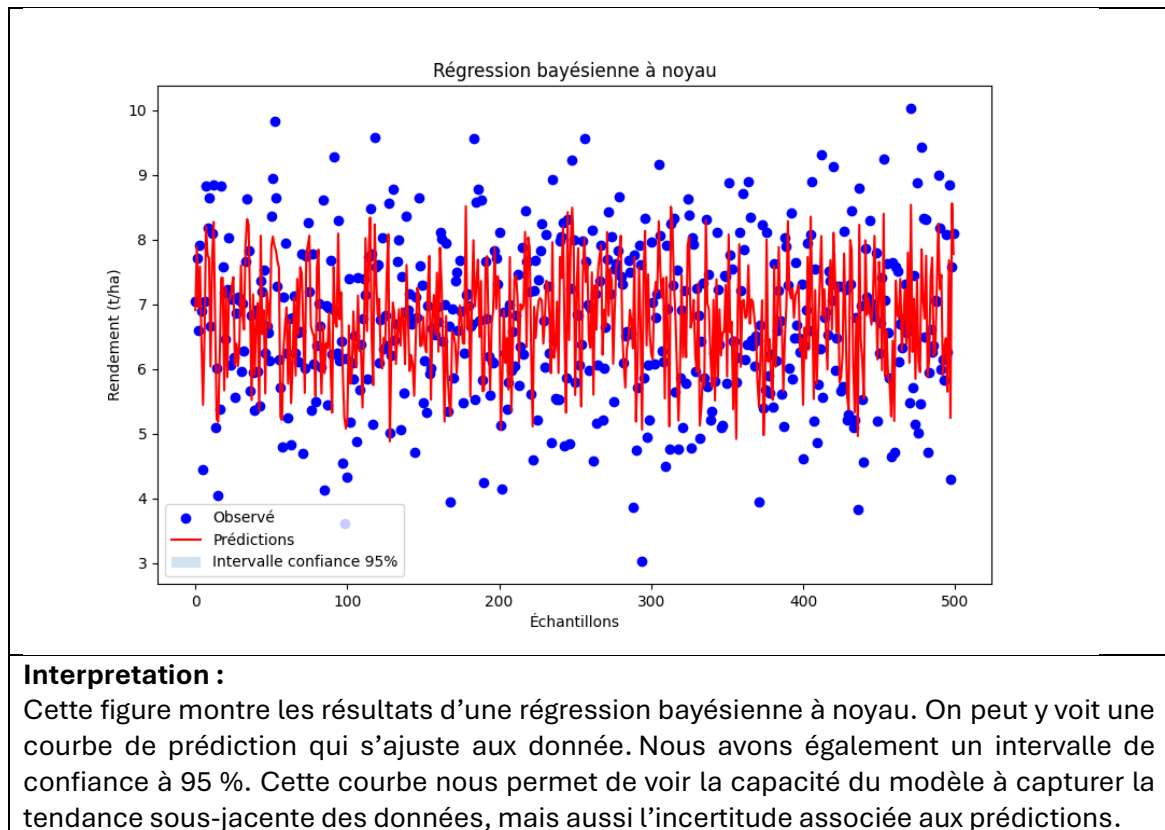
## 10. Distribution a priori et distribution a posteriori

**Distribution a priori :** représente la connaissance initiale sur les paramètres du modèle avant d'observer les données.

**Distribution a posteriori :** représente la mise à jour de nos connaissances après avoir observé les données.

**Exemple appliqué :**

Avant d'obtenir les données sur le rendement agricole, on peut supposer que certaines conditions climatiques auront un effet moyen sur la production (hypothèse distribution a priori). Ensuite après avoir observé les données réelles et leurs impacts sur le rendement, ces connaissances initiales seront ajustées pour refléter précisément le réel des conditions observées nous avons donc notre distribution a posteriori.

**11. Implémentation régression Bayésienne à Noyau****12. Classification bayésienne****Rapport de classification Bayésienne:**

	Metric precision	recall	f1-score	support
Argileux	0.40	0.55	0.46	51
Limoneux	0.37	0.43	0.40	51
Sableux	0.38	0.17	0.23	48
Accuracy			0.39	150
Macro avg	0.38	0.38	0.36	150
weighted avg	0.38	0.39	0.37	150

**Rapport de classification SVM:**

	Metric precision	recall	f1-score	support
Argileux	0.39	0.53	0.45	51
Limoneux	0.38	0.61	0.47	51
Sableux	0	0	0	48
Accuracy			0.39	150
Macro avg	0.26	0.38	0.31	150
weighted avg	0.26	0.39	0.31	150

**Interprétation :**

Les deux modèles affichent une accuracy identique (0.39), mais leurs performances par classe sont différentes. Le modèle bayésien parvient à classer un peu mieux la classe Sableux (f1-score de 0.23 vs 0 pour le SVM). En revanche, le SVM obtient un meilleur recall pour Limoneux, mais échoue complètement pour Sableux, ce qui déséquilibre globalement ses résultats.

En conclusion, même si elles ont une même accuracy (0.39), le modèle bayésien présente des performances plus équilibrées puisqu'il parvient à classer la classe Sableux (même avec un f1-score pas terrible), alors que le SVM échoue complètement pour cette classe.

**13. Incertitude dans les prédictions**

Notre intervalle de confiance est large 95%, cela indique que le modèle est moins sûr de ses prédictions, et donc qu'il y a beaucoup d'incertitude.

**14. Différence entre les noyaux**

Score noyau linéaire : 0.4842134319	Score noyau RBF : 0.48840993
Score noyau polynomial : 0.48958811349	

Dans notre datasheet le noyau impact très peu les performances du modèle. Elles ont un score similaire.

**15. Discussions**

Comme nous avons pu voir précédemment le choix des noyaux n'impacte pas le score du modèle. Nous savons qu'un a priori bien choisi aide à guider l'estimation. Dans notre cas, l'association d'une distribution a priori initial et d'un jeu de données robuste permet d'atténuer l'influence du choix du noyau, c'est pourquoi nous avons quel que soit le noyau le même score. On peut également ajouter que nous avons un dataset plutôt équilibré.