UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

UFR SEN
Sciences Exactes et Naturelles

MÉMOIRE DE RECHERCHE:

PRÉSENTÉ PAR:
GNAGO YANNICK

soutenu le : 18 Mai 2021

L'étude de la fiabilité des prédictions du flux de patients entre un modèle compartimental en épidémiologie (SIR+H) et le processus gaussien .

Le cas des patients infectés par le Covid à l'hôpital universitaire de Nancy.

Université De Reims Champagne-Ardenne
UFR Sciences Exactes et Naturelles
Parcours Statistique pour l'Évaluation et la Prévision.
Master 1 Mathématiques et Applications

ENCADRÉ PAR:
M.BART LAMIROY.

JURY:

M.Philippe Regnault.

ANNÉE UNIVERSITAIRE : 2021-2022

Remerciements

Je souhaite exprimer toute ma gratitude à M. Bart Lamiroy pour son investissement, son attention, sa patience et son enthousiasme tout au long de son suivi pour mon mémoire.

Je remercie M. Philippe Regnault pour avoir accepté de juger et d'être membre du jury. Ce travail de mémoire est une continuité dans le cadre du développement d'un logiciel prédictif du flux des patients nommé FLOWSIM dans un hôpital. Je souhaite ainsi remercier tous les membres de l'équipe de développement pour leurs travaux déjà accomplis. Cela m'a permis véritablement de comprendre l'enjeu du métier de Data scientist que j'aspire faire plus tard.

Plus personnellement, je souhaite exprimer une pensée sincère aux professeurs de ma formation (Mme. Emmanuelle Gautheret , M. Jules Maes, M. Philippe Regnault , M. Amor Keziou) qui m'ont aidé à certains moments quand j'étais en difficulté.

Ce travail de rédaction de mémoire a pu également être mené à son terme grâce de très bons amis qui ont su être présents aux bons moments. Merci à vous.

Je termine ces remerciements par ma famille dont mes essentiels : Ernest (père), Meri (mère). Je ne peux exprimer toute ma gratitude et tous mes sentiments pour votre soutien, votre reconnaissance et votre amour. Merci du fond du cœur.

Résumé

Durant la pandémie du COVID qui a bouleversé le monde entier, les services de santé ont été les premiers à subir d'énormes changements. Dans le but de sauver de monde, plusieurs acteurs ont alors agi à leurs manières pour redonner à une certaine sérénité dans les services de santé. C'est dans cette optique qu'ils se servent des outils (comme l'intelligence artificielle, les statistiques, l'informatique, ...) pour mettre en place des modèles pour prédire le flux des personnes infectées par la maladie

(notée D [1]) afin d'aider au mieux les hôpitaux sur le front de l'épidémie coronavirus. Au regard de cela, M. Bart Lamiroy et son équipe se sont penchés sur la situation de l'unité de soins de réanimation au sein de l'hôpital universitaire de Nancy qui n'était pas en marge de cette crise sanitaire. De ce fait, l'objectif de son équipe étant pareil, i.e., plus précisément de permettre aux dirigeants de cet hôpital de pouvoir estimer le nombre de personnes infectées à court terme (soit sur deux ou trois jours) afin qu'ils prennent rapidement des décisions pour assurer la sérénité dans leurs services.
Notre recherche vise donc d'une part à mettre en place des modèles prédictifs de flux de patients, d'autre part de valider ces modèles en mesurant la qualité des prédictions effectuées afin que les décisions prises soient plus adéquates à la situation réelle de l'hôpital universitaire de Nancy durant cette période.

Tout d'abord, cette partie permet d'expliquer le modèle compartimental en épidémiologie [2] dans notre cas (SIR+H), de donner son utilité en épidémiologie, ensuite de comprendre théoriquement la méthode du processus gaussien, connaître les différents axes de ce processus afin de comprendre son utilité dans nos situations puis je vais programmer cette méthode prédictive en python. Après cela, j'analyserai les prédictions du processus gaussien avec d'autres méthodes comme celle du moindre carrée ou d'autres outils connus afin de pouvoir conclure sur la fiabilité des modèles.

1. D =Disease : la maladie
2. L'épidémiologie est une discipline scientifique qui étudie les problèmes de santé dans les populations, leur fréquence, leur distribution dans le temps et dans l'espace, ainsi que les facteurs exerçant une influence sur la santé.

Table des matières

Table des figures

Chapitre 1

Présentation du mémoire

Durant le contexte sanitaire en 2020, la pandémie a mis en lumière les différents soucis des hôpitaux qui sont entre autres : la pénurie de lits, le manque de personnel, le manque de place dans les unités de soins (de réanimation, intensifs, ...). De nombreux services d'urgence sont en situation de surpopulation. Les unités de soins intensifs ont dû fonctionner à pleine capacité durant cette période. Cela donne lieu à des attentes qui causent de la frustration, de l'anxiété et des conséquences potentiellement nuisibles pour les patients, tout en augmentant la pression sur le personnel.

1.1 Description du problème à résoudre

L'hôpital universitaire de Nancy n'étant pas en marge sur cette situation, les dirigeants avaient du mal à gérer l'augmentation inattendue des flux patients en soins de réanimation. M. Bart Lamiroy et son équipe ont donc décidé de travailler sur des méthodes afin d'améliorer la prise de décisions des dirigeants concernant le flux de patients pour pouvoir mieux s'adapter à cette circonstance sanitaire. Les mesures voulues par les dirigeants étaient alors : de protéger les patients à risque, de réduire la durée du séjour en réanimation, de réorganiser le service de réanimation, de connaître le personnel de santé nécessaire en réanimation. Toutes ces mesures n'étant pas efficaces si on ne peut prédire le flux de patients en réanimation, i.e., avoir une estimation du nombre de patients sur 2 à 3 jours. Cela permettra aux dirigeants de l'hôpital d'anticiper et de s'adapter au mieux aux différentes circonstances liées à cette pandémie. Il en résulte que la méthode de M. Bart Lamiroy et son équipe était donc de pouvoir prédire le flux de patients infectés en réanimation (sur 2 à 3 jours)

afin que l'hôpital puisse prendre rapidement des mesures mieux adaptées.
En effet, il existe plusieurs modèles de prédiction en informatique, parmi celle-ci, nous nous intéressons au modèle SIR (modèle très connu en épidémiologie vu notre contexte) et le processus gaussien (modèle très connu en machine Learning). Nous disposons donc des données représentant l'effectif des patients en réanimation sur cette période de la pandémie.
Décidément, nous sommes confrontés deux problèmes à résoudre :

- Problème 1 : **"Sauver le monde"**, i.e., trouver un outil qui permet de prévoir le flux de patients sur 2 à 5 jours dans les services hospitaliers afin que les dirigeants puissent adopter au mieux des mesures dans le but de sauver plus de vie humaine.
- Problème 2 : évaluer et Mesurer la qualité des modèles prédictifs afin de choisir le meilleur pour notre contexte, i.e. celui avec une faible marge d'erreur afin que cela puisse véritablement aider les dirigeants dans leurs prises de décision durant cette pandémie.

1.2 Hypothèse de solution

Étant donné que notre problème porte sur une maladie infectieuse et vu que le nombre de personnes infectées en réanimation varie au cours du temps, l'ensemble du nombre d'infectés est une distribution normale multivariée [3], supposons qu'il est possible :

- De trouver un modèle compartimental lié à l'épidémiologie (SIR, SEIR,...) (cf. chap.2) .
- De trouver un modèle prédictif du flux d'infectés relative à une distribution gaussienne conjointe (cf. chap.3).
- D'étudier leurs fiabilités avec les méthodes de validation comme le coefficient de détermination (R^2), l'erreur moyenne absolue (MAE), la validation croisée, et autres (cf. chap. 4).

3. Une distribution normale multivariée (gaussienne conjointe) est une généralisation de la distribution normale unidimensionnelle à des dimensions supérieures.

1.3 L'explication de la base de données

La base de données sur laquelle nous allons travailler nous a été fournie par l'hôpital universitaire de Nancy, elle concerne le nombre de patients infectés entrés en soins de réanimation dès le début de pandémie en France. Dans la base, nous avons 2 variables : **la date** considérée comme un entier en prenant le premier jour de la simulation (06/01/2020) égale à 5, i.e. le 5^e jours à partir du 01/01/2020, et **le nombre d'infectées en réanimation**. Vous pourrez aussi remarquer que le nombre d'infectées est en décimale, cela s'explique par le fait que les valeurs ont été corrigées par le facteur 1,5.

	A	B
1		rea
2	55	1.5
3	56	1.5
4	57	1.5
5	58	3
6	59	0
7	60	1.5
8	61	0
9	62	1.5
10	63	0
11	64	1.5
12	65	0
13	66	0
14	67	1.5
15	68	4.5
16	69	25.5
17	70	13.5
18	71	28.5
19	72	12
20	73	30
21	74	30
22	75	15
23	76	19.5
24	77	12
25	78	18
26	79	19.5
27	80	25.5
28	81	18
29	82	12
30	83	10.5

Source : La base de données des entrées en réanimation de l'hôpital de Nancy du 06/01/2020 au 18/03/2020

FIGURE 1.1 – La base de données sur les entrées au mois de mai en unité de réanimation.

Comme énoncé dans la section 1.1, nous expliquerons dans la suite les modèles prédictifs choisies séparément afin que cela soit plus compréhensible avant de les appliquer à notre jeu de donnée.

Chapitre 2

Les modèles prédictifs

Un modèle prédictif est un modèle mathématique et informatique qui permet de probabiliser l'action future d'une situation afin de prendre des décisions, i.e., ils permettent de prévoir l'évolution future (est-ce que tel événement va arriver ?).
Il en existe plusieurs selon le domaine :

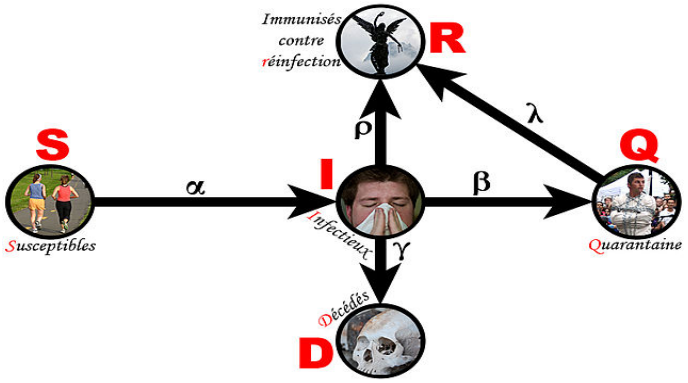
- En informatique : le modèle de régression, de classification,...
- En épidémiologie : le modèle SIR, SEIR,...

Il faudrait aussi reconnaître qu'avec l'essor de l'intelligence artificielle durant cette décennie, plusieurs modèles prédictifs ont été conçus répondant spécifiquement dans les prises de décisions de certaines situations (cf. sur ce site).
Vu notre contexte, nous nous pencherons sur les modèles en épidémiologie et en informatique.

2.1 Les modèles compartimentaux en épidémiologie

Le modèle compartimental en épidémiologie est l'un des modèles les plus utilisés afin de prédire l'évolution d'une épidémie au cours du temps. Son but principal est de guider les dirigeants dans la prise de décision en termes de santé comme dans notre situation. Appelé "modèle à compartiment", i.e. que l'on divise la population en plusieurs catégories.
De façon générale dans une situation d'épidémiologie, on a pour une population donnée (De taille **N**) plusieurs sous-populations (les compartiments) au cours du temps (t), les compartiments basiques sont : **S** pour les personnes saines (susceptible), **I** pour les personnes infectées (Infected), **R** pour les personnes rétablies (Recovered), **D** pour les personnes décédées (Deceased) et **Q** pour les personnes en quarantaine

(Quarantaine) (cf. sur ce site).



Source : Modèles compartimentaux en épidémiologie

FIGURE 2.1 – Exemple d'un modèle avec cinq compartiments, portant les noms de leurs abréviations S, I, Q, R et D. Avec les taux de transmission d'individus allant d'un compartiment dans un autre en lettres grecques minuscules .

2.1.1 Le modèle SIR

Un modèle SIR est défini par : $\mathbf{N} = \mathbf{S}(t) + \mathbf{I}(t) + \mathbf{R}(t)$ représentant la population constante au cours du temps. Il convient de bien différencier les personnes saines des personnes rétablies : les personnes saines n'ont pas encore été touchées par le virus, alors que les personnes rétablies sont guéries, et donc immunisées. Autrement dit, les personnes rétablies ne sont plus prises en compte (cf. ref.[12]).

$$\boxed{S} \xrightarrow{\beta} \boxed{I} \xrightarrow{\gamma} \boxed{R}$$

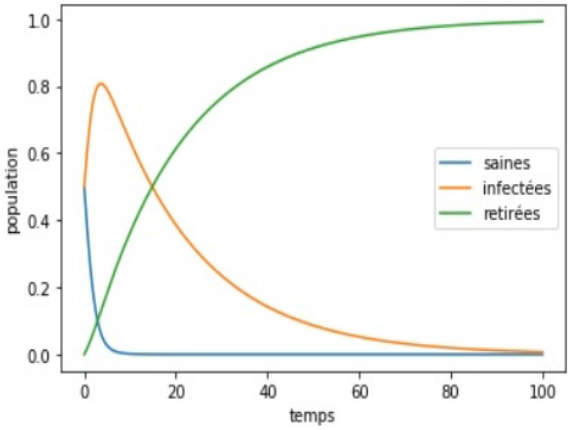
Source Modélisation d'une épidémie, partie 1

FIGURE 2.2 – Représentation d'un modèle SIR, Avec les taux de transmission d'individus allant d'un compartiment dans un autre en lettres grecques minuscules .

Mathématiquement, le modèle SIR est donné par le système suivant :

$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta S(t)I(t) & (1.1) \\ \dfrac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) & (1.2) \\ \dfrac{dR(t)}{dt} = \gamma I(t) & (1.3) \end{cases}$$

Les dérivées permettent de connaître la variation (i.e. si c'est croissant ou décroissant) des fonctions **S**, **I** et **R** en fonction du temps t, afin d'en décrire l'évolution au cours du temps. Le terme **S**(t)**I**(t) représente le nombre de contacts entre des personnes saines et des personnes infectées. β étant le taux de transmission, il y a dès lors β**S**(t)**I**(t) personnes nouvellement infectées. Celles-ci se soustraient des personnes saines (1.1), et s'ajoutent aux personnes infectées (1.2). De même, parmi les personnes infectées, certaines vont guérir : γ étant le taux de guérison, il a γ**I**(t) personnes nouvellement guéries qui s'enlèvent des personnes infectées (1.2) et s'ajoutent aux personnes rétablies (1.3) (cf.ref.[12]).
Ce modèle peut paraître simple, trop simple même, mais il est efficace : il a aidé à la politique sanitaire de vaccination contre la variole au début du XXe siècle. Les personnes saines vaccinées seront automatiquement rétablies si elles sont infectées, et à terme l'épidémie s'arrête.



Source : Modélisation d'une épidémie, partie 1

FIGURE 2.3 – Solution du modèle SIR, Avec Le taux de transmission est de 0,8 et le taux de guérison est de 0,05.

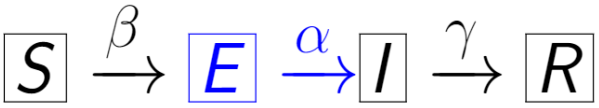
Supposons le cas extrême, i.e., l'épidémie s'arrête, on aura : $\lim_{t\to+\infty} I(t) = 0$, cela prend aussi en compte le cas où la population est vaccinée ou immunisée (immunisé naturelle ou collective). C'est pourquoi, des mesures politiques (le confinement, les mesures barrières dans le cas du COVID) sont prises pour arriver à "diminuer le temps" avant l'arrêt de l'épidémie. Pour cela, il est nécessaire de réduire ce qu'on appelle le taux de reproduction (contagiosité [4]) $R_0 = \frac{\beta}{\gamma}$ (cf. ref.[9]).
Le modèle SIR étant relativement simple, permet d'obtenir une première modélisation d'une épidémie et d'observer l'impact des mesures sanitaires sur son l'évolution.

4. Le taux de reproduction R_0 est le nombre moyen de cas secondaires produits par un individu infectieux au cours de sa période d'infection.

Cependant, celle-ci n'explique pas au mieux notre situation sanitaire vue que durant l'évolution de la pandémie du COVID, l'hôpital universitaire de Nancy a dû mettre en place plusieurs mesures qui ont sûrement influencé le modèle SIR. Il faut aussi remarquer qu'une nouvelle sous-population sera ajoutée : les personnes infectées non-infectieuses (**Exposed**) car celle-ci est très importante dans notre contexte vu que ces personnes exposées (**Exposed**) ne sont pas contagieuses, mais elles pourront contaminer les personnes saines (**Susceptible**). La prochaine partie portera donc sur l'explication d'un modèle mieux adapté à notre contexte qu'on appellera le modèle SIR+H.

2.1.2 Le modèle SIR+H (le modèle adapté à notre contexte)

Initialement, notre modèle était constitué des paramètres suivants : le nombre de personnes susceptibles d'être infectées (**S**), le nombre d'infectées (**I**), les personnes rétablies (**R**) . Le nouveau paramètre qui sera ajouté est : le nombre de personnes infectées non-infectieuses qui pourront transmettre le virus (**E**) (cf. sur ce site).
Cela nous permettra de prendre en compte la durée d'incubation du virus (ici pour le COVID entre 3 et 14 jours selon le variant) via le taux d'incubation (α).
Pour ce type de modèle communément appelé modèle SEIR, il est donc primordial de calculer les paramètres : α (le taux d'incubation), β (le taux de transmission) et γ (le taux de guérison) (cf. ref.[9]). L'évaluation de ces paramètres se fait généralement en collaborant avec les équipes médicales, les épidémiologistes et les virologistes (cf. section.2.1.3).



Source : Modélisation d'une épidémie, partie 2.

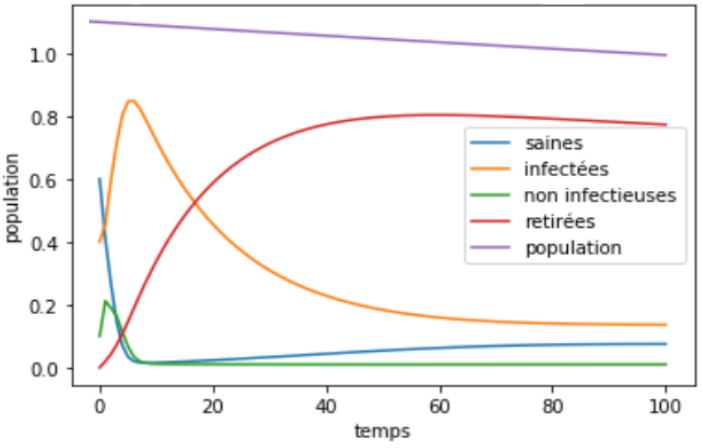
FIGURE 2.4 – Représentation d'un modèle SEIR.

Mathématiquement, le modèle SEIR est donné par le système suivant :

$$\begin{cases} \frac{dS(t)}{dt} &= -\beta S(t)I(t) \\ \frac{dE(t)}{dt} &= \beta S(t)I(t) - \alpha E(t) \\ \frac{dI(t)}{dt} &= \alpha E(t) - \gamma I(t) \\ \frac{dR(t)}{dt} &= \gamma I(t) \end{cases}$$

Avec :
-**S**(t) : l'effectif de personnes sensibles le jour t.
-**E**(t) : l'effectif de personnes exposées le jour t.
-**I**(t) : l'effectif de personnes infectées le jour t.
-**R**(t) : l'effectif de personnes guéries le jour t.
-α : le taux d'incubation du virus.
-β : le taux de transmission par jour.
-γ : le taux de rétablissement par jour.

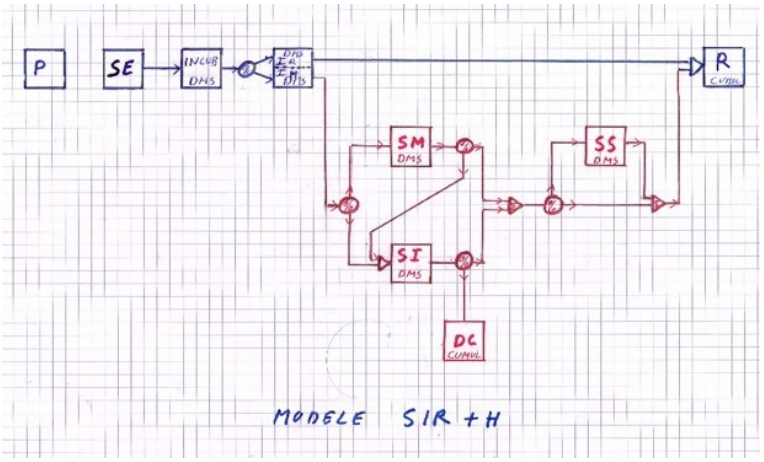
Dans l'annexe à la page 49, nous avons mis au point un modèle SEIR python simple pour mieux le comprendre.



Source : Modélisation d'une épidémie, partie 2.

FIGURE 2.5 – Exemple d'un modèle SEIR avec α=0.75 , γ=0.05 et β=0.8 .

Tout en tenant compte les unités de soins de l'hôpital universitaire de Nancy, nous avons alors recontextualisé en créant des compartiments relatifs aux unités de soins de l'hôpital afin d'avoir le modèle plus adapté (le modèle SIR+H).



Source : D'après l'étude effectuée du professeur M. Lamiroy avec son équipe en tenant compte du fonctionnement de l'hôpital universitaire de Nancy.

FIGURE 2.6 – Graphe du modèle SIR+H

Ce nouveau modèle représente un peu plus notre situation sanitaire au sein de l'hôpital (cf. ref.[1]). Nous avons alors :

- P : la population saine (les patients non infectés par le COVID).

- SE : les patients exposés.

- IN : les patients en phase d'incubation.

- IR : les incubés rétablies.

- IH : les incubés hospitalisés.

- I : les personnes infectées (par le COVID).

- R : les patients rétablis.

- SM : soins Médicaux (DMS : durée moyenne de séjour).

- SI : soins Intensifs (DMS).

- SS : soins de Suite (DMS).

- DC : les personnes décédées par le virus.

De plus, durant la crise sanitaire liée au COVID en FRANCE, il y'a plusieurs phases de propagation (cf. sur ce site) :

- Phase 1(Avant le 17 mars 2020) : début de la crise sanitaire et propagation du virus.

- Du 17 mars au 11 mai 2020 : Premier confinement national.

- Phase 2(Du 11 mai au 30 octobre 2020).

- Du 30 octobre au 15 décembre 2020 : Deuxième confinement.

- Phase 3(Du 15 décembre 2020 au 3 avril 2021).

- Du 3 avril au 3 mai 2021 : Troisième confinement.

Cela est très important pour notre situation au sein des unités de soins de l'hôpital universitaire de Nancy, ainsi dans la prochaine partie, l'évaluation des paramètres du modèle se fera alors en tenant compte des phases de propagation du virus en France. Dans la section. ref2.1.3, nous décrivons la façon dont ces paramètres du modèle SIR+H sont évalués.

2.1.3 L'estimation des paramètres

Considérant les informations ci-dessus, les paramètres que nous évaluerons pour la modélisation SIR+H à chaque phase de la propagation du virus dans le service de santé sont :
- population : la taille de la population du modèle SIR+H (fixe).
- lim_time : le nombre d'itérations de simulation à exécuter (fixe).
- R_0 : le taux de reproduction du virus à chaque phase.
- beta (β)=R_0/dm_r : taux de transmission initial (variable en fonction des phases).
- patient0 : le nombre d'infectés au début de la simulation (nombre fixe de cas infectés du modèle SIR+H au début de la simulation).
- pc : le pourcentage de transmission entre chaque compartiment (observé entre les unités de soins).
- dm_incub : durée moyenne d'incubation du virus(α).
- dm_r : durée moyenne de rétablissement (γ).

Le R_0 est calculé avec la base de données des indicateurs en faisant les moyennes de la colonne étiquette « R » sur les périodes concernées, et relatives au département de la Meurthe-et-Moselle (cf. sur ce site).
Quant aux paramètres (pc, dm), ils sont déterminés en calculant les durées moyennes ou les probabilités de transition du modèle SIR+H.
En fait, tout au long de l'évolution du modèle, ces paramètres (beta, beta_post, patient0, dm_r, dm_incub) seront estimés au fur à mesure et sauvegarder dans le fichier ".json" pour être réutilisé. Le modèle sera alors défini comme une fonction prédictive :$f(beta, beta\_post, patient0, dm\_r, dm\_incub, t)$.
Ainsi pour pouvoir estimer les paramètres, on utilise l'algorithme de Levenberg-Marquardt qui permettra d'obtenir une solution au problème de minimisation de la fonction prédictive non-linéaire. Cet algorithme est basé sur l'algorithme du gradient et de Gauss-Newton et plus stable dans le cadre d'un problème de minimisation d'une fonction non-linéaire avec plusieurs variables comme notre situation (cf. ref

[7]).
Dans l'annexe à la page 48, nous expliquons en détail l'algorithme qui permet d'estimer ces paramètres.

Conclusion : Le modèle SIR+H trouvé est bien un modèle prédictif, il suffit juste de bien choisir les données (SE, IN, IR, IH, SM, SI, SS, R) tenant compte de la situation réelle dans les unités de soins de l'hôpital universitaire de Nancy afin de prédire au mieux le nombre d'infectés.
Comme remarque, les prédictions vont fortement dépendre de l'estimation des paramètres et de la concordance des données recueillies instantanément dans les unités de soins. Par ailleurs, comme signifié précédemment, nous étudierons dans la suite un autre modèle prédictif basé le fait que le nombre d'infectés suit une distribution gaussienne conjointe au cours du temps.

2.2 Le processus gaussien

Un processus gaussien (GP) est une collection de variables aléatoires, dont un nombre fini quelconque a une distribution gaussienne conjointe. Un GP peut être interprété de manière flexible comme une distribution sur des fonctions où les variables aléatoires sont les valeurs de fonction d'une fonction latente $f(x)$ (où x est le vecteur d'entrée) (cf. ref [11].[6]).
Nous considérons donc un ensemble de données d'apprentissage :
$D = \{(x_i, y_i)\}_{i=0}^{N}$ avec $y_i \sim f(x_i)\}$. Avec $f(x) \sim GP(m(x), K(x, x'))$
avec $m(x)$: fonction moyenne et $K(x, x')$: fonction de noyau (covariance),

$$m(x) = E[f(x)]$$

$$K(x, x') = E[(f(x) - m(x))(f(x') - m(x'))]$$

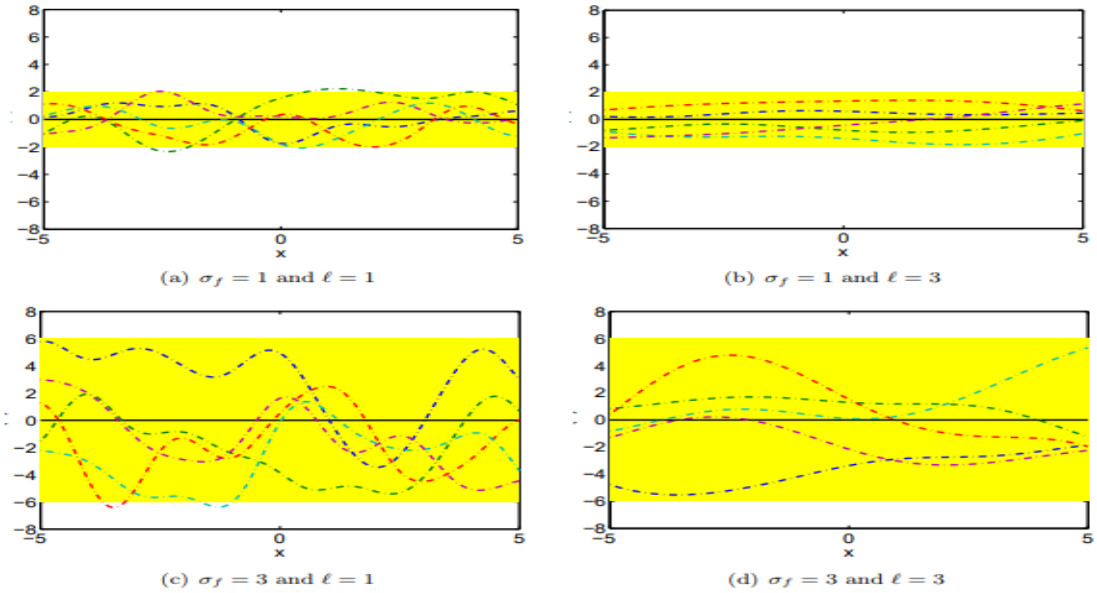
D'après la définition d'un GP, la fonction $f(x) = \{f(x_1), \ldots, f(x_n)\}$ des vecteurs d'entrée $x = \{x_1, \ldots, x_n\}$ est une distribuée de manière gaussienne, i.e. $\mathcal{N}(m, K)$ avec le vecteur moyen $m = \begin{pmatrix} m(x_1) \\ \ldots \\ m(x_n) \end{pmatrix}$ et $K = \begin{pmatrix} K(x_1, x_1) & \ldots K(x_1, x_n) \\ \ldots & \ldots \quad \ldots \\ K(x_n, x_1) & \ldots K(x_n, x_n) \end{pmatrix}$.
Concernant le vecteur moyen m, nous utiliserons un GP de moyenne nulle.
Dans ce cas : $f(x) \sim GP(m(x) = 0, K(x, x') = K)$.

2.2.1 La fonction de covariance

Concernant la fonction de covariance K, ils existent plusieurs façons de la définir à condition qu'elle soit une matrice symétrique et semi-définie positives. Elle joue un rôle important, car elle permet de voir la similarité des valeurs de la fonction aux différentes entrées ; en d'autres termes, c'est la matrice d'interactions entre les entrées d'apprentissage x et x'.
La matrice de covariance permet aussi d'établir une relation lisse entre les données, i.e., la prédiction se fera non loin des données d'apprentissage. C'est donc un avantage du processus gaussien, car il se fait sur tout l'ensemble des données (cf. ref.[4]).
Dans notre situation, il s'agira de définir la fonction du noyau grâce aux premières données avec le modèle SIR+H afin de définir la fonction covariance.
Il existe donc plusieurs façons de la définir.

- L'exponentielle au carré (SE) : la fonction la plus utilisée dans l'apprentissage automatique.
$$k(x, x') = \sigma^2 \exp -\frac{(x - x')^T (x - x')}{2l^2},$$
où l'hyperparamètre σ contrôle l'amplitude l contrôle l'échelle de longueur de nos données. La figure 2.7 montre l'influence de σ et de l, les fonctions deviennent plus plates lorsque l est plus grand, et l'amplitude de la fonction devient plus grande lorsque σ devient plus grand. On peut remarquer que la fonction est donc stationnaire.
Il y a aussi d'autres fonctions non-stationnaires pour définir $k(x, x')$:
- Le produit scalaire : $k(x, x') = l^2 x^T x'$.
- Le réseau neuronal : $k(x, x') = \sigma^2 \sin^{-1}(\frac{x^T x'}{l^2 \sqrt{(1+\frac{1}{l^2} x^T x)(1+\frac{1}{l^2} x'^T x')}})$.

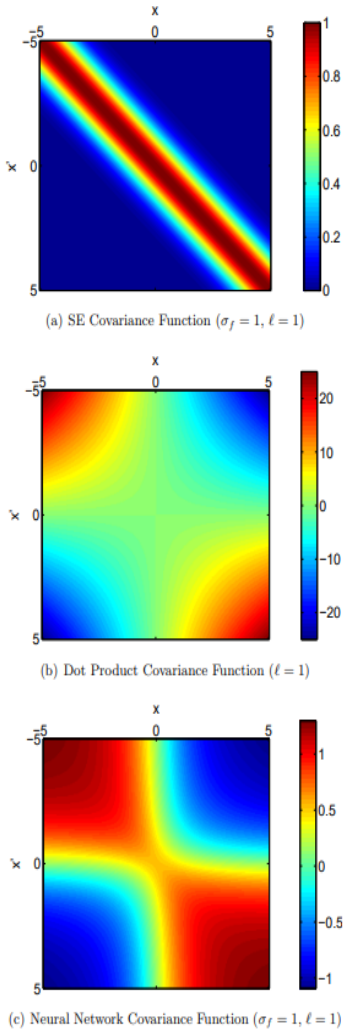
Sur la figure 2.8, on observe que les fonctions de covariance non-stationnaires ne sont pas seulement liées à $x - x'$, mais aussi aux emplacements d'entrée x et x' eux-mêmes. Cela c'est pourquoi elles sont souvent utilisées pour capturer les corrélations non stationnaires.



Source : cf. ref [11], page 7.

FIGURE 2.7 – L'influence de σ et l pour un GP avec la SE.

Pour notre contexte, nous avons choisi l'exponentielle au carré (SE) vu que dans notre donnée, à chaque jour l'on a associé un unique chiffre représentant le nombre d'infectées en réanimation (cf. section.1.3).



Source : cf. ref [11], page 9.

FIGURE 2.8 – La comparaison de la stationnarité de la matrice de covariance utilisant différentes fonctions de covariance.

2.2.2 La régression par le processus gaussien

Supposons que la procédure de génération de données soit basée sur le modèle d'observation suivant,

$$y = f(x) + \epsilon, \qquad (a)$$

Où le vecteur d'entrée est $x \in R^d$, le vecteur de sortie est $y \in R$ et le bruit d'observation est $\epsilon \sim N(0, \sigma_y^2)$ (hypothèse d'homoscédasticité).
En général, il y a deux tâches dans un problème de régression non-linéaire.

1. Apprendre les paramètres du modèle à partir d'un ensemble d'apprentissage $(X, y) = \{(x_i, y_i)\}_{i=1}^N$.

2. Prédire $f(X_*)$ pour un ensemble d'entrées de test donné $X_* = \{x_*^i\}_{i=1}^M$ avec $M > N$.

Le processus gaussien (GP) est l'un des plus populaires modèles de régression non-paramétrique, i.e. La fonction d'estimation ou prédicteur ($yi \sim f(x_i) = a_0 + \sum a_i * K * (x - x_i)$) ne prend pas de forme prédéterminée, mais elle construite selon des informations des données. Cela nous permet de faire une inférence bayésienne analytiquement traçable pour obtenir une distribution prédictive sur $f(X_*)$, ce qui n'est pas le cas pour la régression linéaire bayésienne (cf ref.[3]).
Pour simplifier le processus gaussien, on suppose que $f(x) \sim GP(0, K(x, x'))$ avec les hyperparamètres σ, l pour la fonction de covariance et σ_y^2 pour la variance du bruit. Nous utilisons d'abord l'ensemble d'apprentissage (X, y) pour déduire σ, l et σ_y^2, qui sont les paramètres du modèle (cf. ref.[4]).
Alors la vraie question est de savoir comment trouver ces hyperparamètres de façon à ce que le processus gaussien suive au mieux l'évolution de nos données.
Nous expliquerons dans l'annexe à la page 54 la manière dont les hyperparamètres sont trouvés.

2.2.3 La prédiction

Après avoir défini les hyperparamètres, nous pourrons maintenant trouver la prédiction de $f(X_*)$ pour l'entrée X_*. Plutôt de se contenter d'une prédiction ponctuelle de $f(X_*)$, un GP permet d'obtenir une distribution prédictive bayésienne complète sur $f(X_*)$. En somme, le principe est qu'on va chercher à exploiter ce qu'on connaît déjà, donc l'ensemble des événements précédemment observés, pour inférer la probabilité des événements que nous n'avons pas encore observés.

En effet, vu notre contexte, le modèle prédictif du processus gaussien sera comme une amélioration du premier modèle prédictif (SIR+H) effectué sur un certain temps d'observation qui augmentera au fur à mesure, puis la prédiction avec GP permettra de déduire le nombre d'infectés sur les prochains temps d'observation.

Concrètement, nous pouvons d'abord tirer parti de $f(X) \sim GP(0, K(x, x'))$ et du modèle d'observation gaussien dans l'équation (a) pour obtenir que la distribution conjointe qui est gaussienne :

$$\begin{pmatrix} f(X) \\ f(X_*) \end{pmatrix} \sim N(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(X,X) + \sigma_y^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{pmatrix}) \quad avec \quad \epsilon \sim N(0, \sigma_y^2)$$

En se basant sur cette distribution conjointe et la propriété d'une distribution gaussienne, on obtient que la distribution prédictive $f(X_*)$ est aussi gaussienne avec comme moyenne $\hat{f}(X_*)$ et la covariance $\hat{\sum} f(X_*)$:

$$\hat{f}(X_*) = K(X_*, X)[K(X, X) + \sigma_y^2 I]^{-1} y,$$

$$\hat{\sum} f(X_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_y^2 I]^{-1} K(X_*, X)^T,$$

où $K(X_*, X)$ désigne une matrice de covariance M × N dans laquelle chaque entrée est calculée par la fonction de covariance $K(x, x')$ avec l'apprentissage des hyperparamètres θ déjà évaluée. Aussi $K(X, X)$ et $K(X_*, X_*)$ sont construits de manière similaire.
Nous avons réalisé un exemple de l'algorithme de prédiction de processus gaussien en annexe à la page 52.

2.3 Bilan récapitulatif

Dans ce chapitre, nous avons expliqué séparément sur les deux modèles prédictifs qu'on utilise pour prédire le nombre d'infectes en soin de réanimation afin que l'hôpital puisse bien s'organiser face à la pandémie qui bouleverse leurs unités de soins.
D'une part, nous avons le modèle épidémiologique (SIR+H) développé par nous en se basant sur le modèle SEIR et en tenant compte du fonctionnement les hôpitaux universitaires de Nancy et des phases de propagation du virus COVID en France. Nous avons vu que ce modèle dépend fortement de la détermination de certains paramètres liés à l'évolution du virus.
D'un autre côté, nous avons le procédé GP, qui est un algorithme d'apprentissage automatique appliqué pour traiter un problème de régression essentiellement basé sur le comportement des données. Nous avons pu voir qu'il dépend aussi fortement des hyperparamètres. En fait, c'est un processus d'optimisation qui, basé sur une série de données d'apprentissage, définit l'évolution de ces données par une fonction et permet de prévoir les données futures.

C'est au regard de cela que le GP sera idéal pour notre contexte vu que nos prédictions vont tenir compte du comportement antérieur de l'évolution du virus au sein de l'unité de réanimation en effet le processus GP apprendra du modèle (SIR+H) afin de connaître l'évolution des nombres d'infectés en soins de réanimation sur un certain temps d'observation pour finalement prédire cette évolution pour les jours à venir.

Dans le chapitre suivant, nous appliquerons ces deux modèles de prédiction à notre ensemble de données afin d'analyser les résultats.

Chapitre 3

Application des modéles

Cette partie est totalement consacrée à l'application des modèles de prévision décrits ci-dessus.

3.1 SIR+H : l'estimation à partir de données réelles.

Comme on l'a vu précédemment, la prévision par le modèle SIR dépendra de l'estimation des paramètres (beta, beta_post, patient0, dm_r, dm_incub) ce qui se fera progressivement en prenant en compte les paramètres trouvés antérieurement. Ces paramètres seront automatiquement sauvegardés dans des fichiers que le modèle réutilisera pour des prédictions futures. Pour exemple, voici l'image du fichier contenant les premières estimations des paramètres trouvées après compilation :

```
[[Model]]
    Model(fitter)
[[Fit Statistics]]
    # fitting method   = Nelder-Mead
    # function evals   = 622
    # data points      = 5
    # variables        = 5
    chi-square         = 5.16318845
    reduced chi-square = 5.16318845
    Akaike info crit   = 10.1605820
    Bayesian info crit = 8.20777153
##  Warning: uncertainties could not be estimated:
    this fitting method does not natively calculate uncertainties
    and numdifftools is not installed for lmfit to do this. Use
    `pip install numdifftools` for lmfit to estimate uncertainties
    with this fitting method.
[[Variables]]
    beta:       0.36407030 (init = 0.3333333)
    beta_post:  0.12414414 (init = 0.06666667)
    patient0:   13.9695328 (init = 15)
    dm_r:       6.00000000 (init = 9)
    dm_incub:   4.44170872 (init = 4)

Optimal values : {'beta': 0.36407030464863793, 'beta_post': 0.12414414108721955,
'patient0': 13.969532841037196, 'dm_r': 6.000000001222915, 'dm_incub':
4.4417087204865355}
```

Source : Extrait d'un fichier ".json" après l'application avec le modèle SIR+H.

FIGURE 3.1 – Le fichier de la première estimation des paramètres

Pour les prédictions, nous fixons un certain nombre d'observations (ici les 3 pre-

mières) en estimant au fur à mesure que les paramètres précèdent pour prédire le nombre d'infectes sur les jours restants.
Sur la figure 3.2, on observe que les prédictions trouvées sont très vite dégénérées, elles n'avoisinent pas le nombre réel d'infectés en soin de réanimation. On pourrait remarquer que les prédictions sont très influencées par le choix des paramètres (beta, beta_post, patient0, dm_r, dm_incub) qu'on fixe dès le départ (cf. ref.[2]).

	A	B	C	D	E	F	G	H	I	J
1		date	value	predictions_5	predictions_6	predictions_7	predictions_8	predictions_9	predictions_10	predictions_11
2	0	55.0	1.5	0	0	0	0	0	0	0
3	1	56.0	1.5	0	0	0	0	0	0	0
4	2	57.0	1.5	0	0	0	0	0	0	0
5	3	58.0	3.0	0.004884678525944548	0.004884678525944548	0.004508934023948814	0.004508934023948814	0.006011912031931753	0.004508934023948814	0.006011912031931753
6	4	59.0	0.0	0.009878333075999862	0.009878333075999862	0.0091184613009229 5	0.0091184613009229 5	0.012157948401230599	0.0091184613009229 5	0.012157948401230599
7	5	60.0	1.5	0.0099315929500662 8	0.0099315929500662 8	0.009167624261599642	0.009167624261599642	0.012223499015466191	0.009167624261599642	0.012223499015466191
8	6	61.0	0.0	0.009706162803768 51	0.009706162803768 51	0.008959534895786317	0.008959534895786317	0.011946046527715087	0.008959534895786317	0.011946046527715087
9	7	62.0	1.5	0.010180474307027444	0.010066470583819 38	0.009362493066581556	0.009366059710546663	0.012398820956565 45	0.009340130600403711	0.012385161415104881
10	8	63.0	0.0	0.012186489807581 31	0.011773568131798143	0.011115965586997493	0.011128256983353 78	0.014496772645730668	0.011028179982770758	0.01444044542710897
11	9	64.0	1.5	0.014626375705115211	0.013923817353317007	0.013254107438314 74	0.013273169902096652	0.017064045446985032	0.013084215892880367	0.016947550978504516
12	10	65.0	0.0	0.016505305235295 24	0.015660746197334992	0.014899290398703 85	0.014918831912330965	0.019028805696533 53	0.014656371547647403	0.018849999122082746
13	11	66.0	0.0	0.018012759543751405	0.017108727605333267	0.016212429720364428	0.016228952102546 12	0.020575583286300513	0.015899410037064493	0.02033172940569889
14	12	67.0	1.5	0.019633433813058 84	0.018629713808973828	0.017605509639987	0.017619774090891 35	0.022173724389903 31	0.017204643758626268	0.021851928667890888
15	13	68.0	4.5	0.021721841754016903	0.020513170918720 06	0.019382066335302442	0.019396515253229628	0.024172468839061753	0.018859644565773034	0.02374964795307352
16	14	69.0	25.5	0.024191439572626 53	0.022729355588178624	0.021473221661271 44	0.021487695401749 37	0.026502290537990865	0.020799591929127 83	0.025954933239502 18
17	15	70.0	13.5	0.026821103088068468	0.025133793288521226	0.023694333687155 79	0.023705800456086 57	0.028959954421167028	0.022850260508100 1	0.028268838792035435
18	16	71.0	28.5	0.029546670081820468	0.027675736907890717	0.025988700483335677	0.025993766839748 7	0.031477152598135 91	0.024956642267043763	0.030624248821879578
19	17	72.0	12.0	0.032465727658122 09	0.030412757294795402	0.028432109447217625	0.028428795984899 87	0.034125810098962 72	0.027186301548156 74	0.033089279988365 76
20	18	73.0	30.0	0.035707554137546 09	0.033435048920863304	0.031127942011455884	0.031115336701659734	0.037009747714202 58	0.029632413583505 93	0.035761985751554 88
21	19	74.0	30.0	0.039323379470928 56	0.036792647469640 13	0.034117834075251 71	0.034094564564594 07	0.040171334204778 85	0.032331581677980545	0.038680743915453945
22	20	75.0	15.0	0.043303320483563 68	0.040497426098996 38	0.037393880401663 06	0.037357350983085 28	0.043601655265272335	0.035274853614095 6	0.041834615558090 11
23	21	76.0	19.5	0.047644473141247284	0.044560191620690 97	0.040952119711420334	0.040898920774484 84	0.047292926814060 22	0.038456202552583 95	0.045213723466769086
24	22	77.0	12.0	0.052385102547372 37	0.049013313493199 04	0.044819908453547 17	0.044746528128862 72	0.051266039690686 06	0.041897249682215 52	0.048853311846757 64
25	23	78.0	18.0	0.058311532444368 33	0.054628805281160 25	0.049711417666584 2	0.049614423449295 83	0.056449913627628 72	0.046304008722181 95	0.053623283233496015
26	24	79.0	19.5	0.064190570235772 95	0.060165642455227444	0.054477574807928 67	0.054353238140440234	0.061290496047171 71	0.050519286266300 16	0.058018194352804 39
27	25	80.0	25.5	0.069853207933018 61	0.065456146782790 44	0.058953550892706 91	0.058797439009498 02	0.065549751383231 05	0.054370895784943 83	0.061776725978155 25
28	26	81.0	18.0	0.076658430985876 16	0.071862670880676 04	0.064384617404020 59	0.064191494149473 4	0.070869440880902 02	0.059096440233724 97	0.066535156649244 31
29	27	82.0	12.0	0.084304666080008 83	0.079065855485520 07	0.070475478150049 02	0.070239949290409 24	0.076820154775632 03	0.064388105594309 56	0.071855848971390 27
30	28	83.0	10.5	0.092839844166334 59	0.087088762192797 72	0.077255724452940 77	0.076972406030497 01	0.083413602059608 54	0.070265551599377 77	0.077745342517081 68

Source : Extrait des fichiers ".csv" générés pour les prédictions du modèle SIR+H.

FIGURE 3.2 – Les prédictions évaluées du modèle SIR+H.

3.2 Amélioration avec le processus gaussien

Comme indiqué dans la section 2.3, le processus gaussien sera utilisé pour une amélioration du modèle (SIR+H) effectué précédemment.
Dans le but d'être un peu plus précis, nous fixons les cinq premières observations puis en rajoutant au fur à mesure que la prochaine valeur prédite par le modèle afin de

prédire le nombre d'infectes sur les jours restants. En fait cette technique s'assimile plus à la technique du "train-test split" [5] de la méthode de validation croisée (cf. annexe. page54).

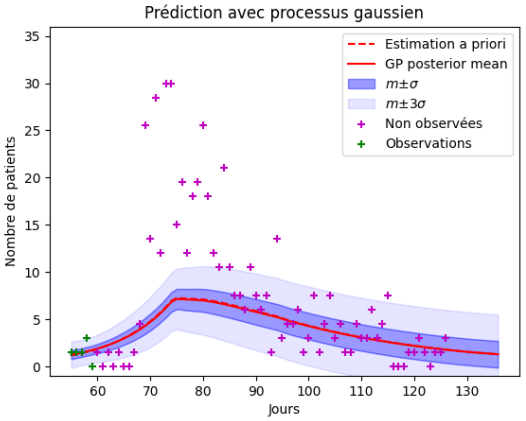
	B	C	D	E	F	G	H	I	J	K
1	date	value	predictions_5	predictions_6	predictions_7	predictions_8	predictions_9	predictions_10	predictions_11	predictions_12
2	55.0	1.5								
3	56.0	1.5								
4	57.0	1.5								
5	58.0	3.0								
6	59.0	0.0								
7	60.0	1.5	1.9067407613119058							
8	61.0	0.0	2.0795207939408336	1.2695997787051851						
9	62.0	1.5	2.270510941941369	1.2324195751490628	1.0643480150914384					
10	63.0	0.0	2.481546960328734	1.2136238658772092	1.1183871831605003	1.699050734653132				
11	64.0	1.5	2.7146280307681376	1.216960160685503	1.2388498569623645	1.8272618405149346	1.3680790088060673			
12	65.0	0.0	2.97193113307156	1.24634570830296	1.423784784436836	1.971269150703304	1.4412489914089746	1.5079033857287296		
13	66.0	0.0	3.2558263597108166	1.3058572750301896	1.665661701458418	2.1326364973298815	1.5258589850889546	1.6071023566476734	1.1930704590600492	
14	67.0	1.5	3.568893617059568	1.3997229024503608	1.9537248105660834	2.3130334647845925	1.6230077135778358	1.720425011866859	1.2448447116018662	-0.004035941595437187
15	68.0	4.5	3.9139410281721214	1.5323165680217512	2.2765335696079156	2.514243754817698	1.7338333740490155	1.8492051198813073	1.3075441535606462	-0.1464571372019925
16	69.0	25.5	4.294024941991179	1.708155876763867	2.62412003025016	2.7381746985456528	1.8595156366530805	1.9948430296359891	1.3821584419353572	-0.27511257664364197
17	70.0	13.5	4.712471785509373	1.931903677815129	2.9894032634197107	2.986868167457685	2.0012787547379203	2.1588111142422837	1.4696921392533955	-0.3869665645963485
18	71.0	28.5	5.172901235289773	2.208372918775378	3.3687598260460603	3.262512448060853	2.1603951953412004	2.3426599675808726	1.5711654104565635	-0.47907035573011836
19	72.0	12.0	5.6792505451499915	2.5425346597764698	3.761870466788989	3.5674549504055046	2.338189561934033	2.548025221098131	1.6876150451763017	-0.5485934701169741
20	73.0	30.0	6.2358011960103905	2.9395318956302017	4.171094206995173	3.904216811100584	2.5360441787389356	2.776635999587755	1.8200970401256356	-0.5928500191574648
21	74.0	30.0	6.847207278868939	3.4046981704615424	4.600647440628446	4.275508919939449	2.755405683103147	3.0303245633765625	1.969690162754734	-0.6093200964174792
22	75.0	15.0	7.156153026096076	3.5570786965296075	4.807711979929509	4.428300025290249	2.842861637273112	3.1199831497153716	2.00997465453298	-0.6984257504965421
23	76.0	19.5	7.1251560757703926	3.3564231631232206	4.769703217846435	4.336602158269484	2.7839113903050574	3.026767743991161	1.929323092243148	-0.8685551942796743
24	77.0	12.0	7.087743467952621	3.1579195464952923	4.718007321546033	4.23662585007895	2.7229988453464946	2.927666278424923	1.8469877787204916	-1.0229086488930323
25	78.0	18.0	7.063770634458736	2.9821235220927274	4.667885603724573	4.141940385546709	2.668476474893061	2.832748663672909	1.7697589905180242	-1.1553379322863813
26	79.0	19.5	7.0463386900124405	2.8285137267554394	4.617661464914789	4.050671242225826	2.6184190174532196	2.7413238826540707	1.6969880778043704	-1.2662552058371799
27	80.0	25.5	7.004001131774543	2.6769105174878716	4.550204715724104	3.9469342424150913	2.5615954782319523	2.643006135002778	1.6213717354086694	-1.3620653670802003
28	81.0	18.0	6.927083167558893	2.520564801357974	4.460814091226998	3.8262545890059783	2.494665681998629	2.5350188403244163	1.5409334416487315	-1.4448773242805921
29	82.0	12.0	6.8277948177656445	2.367249488873986	4.35744081913722	3.695713298445491	2.422224166324144	2.4223053735597415	1.4590507906820307	-1.512367097209801
30	83.0	10.5	6.716138185165724	2.223134765343044	4.246501510854192	3.561043419070373	2.347974387066189	2.308821511544546	1.3783923160761207	-1.5629410620342772

Source : Extrait des fichiers ".csv" générés pour les prédictions du modèle GP.

FIGURE 3.3 – Les prédictions évaluées du modèle GP.

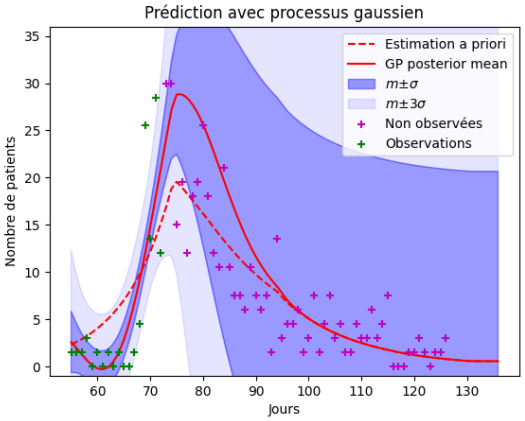
Il faut noter que sur la première prédiction (cf.figure. 3.4), le GP coïncident très bien avec l'estimation a priori de nos données réelles et la région de confiance des prédictions est très régulière, mais cela commence réellement à se décaler dès la 10^e prédictions où que le GP ne coïncide plus avec l'estimation des données réelles et l'intervalle de confiance devient instable.

5. L'approche Train-Test Split consiste à décomposer de manière aléatoire un ensemble de données. Une partie servira à l'entraînement du modèle de Machine Learning, l'autre partie permettra de le tester pour la validation.



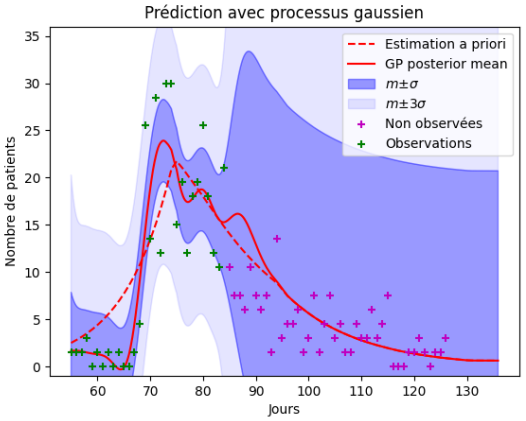
Source : Extrait après le lancement du package des modèles de prédiction.

FIGURE 3.4 – La 1^{er} prédiction évaluée du modèle GP



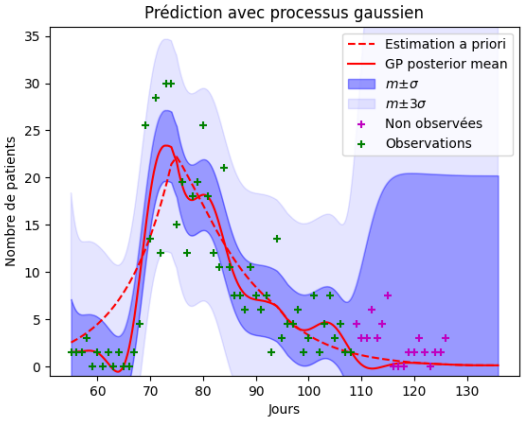
Source : Extrait après le lancement du package des modèles de prédiction.

FIGURE 3.5 – La 14^{e} prédiction évaluée du modèle GP



Source : Extrait après le lancement du package des modèles de prédiction.

FIGURE 3.6 – La 26^{e} prédiction évaluée du modèle GP



Source : Extrait après le lancement du package des modèles de prédiction.

FIGURE 3.7 – La 50^{e} prédiction évaluée du modèle GP

Décidément, nous constatons que les prédictions du modèle SIR+H avec l'amélioration du GP semblent concorder à l'estimation à priori des vraies valeurs sur les premières instants, en fait cet aspect paraît répondre parfaitement aux problèmes 1 et 2 énoncés dans la section 1.1, i.e. trouver un bon outil de prédiction du nombre d'infecté à court terme afin que les dirigeants puissent adopter rapidement des mesures dans le but de sauver le plus de vie humaine ; cependant ce modèle développé est-il fiable, i.e. les prédictions trouvées à court terme avoisinent-elles le nombre réel

d'infectés dans notre situation ?

Chapitre 4

Analyse des prédictions des modéles.

Nous proposons ici des outils d'analyse prédictive pour vérifier la fiabilité de nos deux modèles développés.Avant de répondre à cette problématique, nous expliquerons ses différents outils dans le cadre de la régression.

4.1 les outils d'évaluation des modèles de prédictions

En effet, ces outils souvent appelés métriques en machine Learning dans le but d'évaluer des modèles de prédictions. Il existe plusieurs métriques mais nous expliquerons juste ceux que nous avons pu utiliser dans le cadre de notre travail.

4.1.1 Mesures de l'erreur de prédiction(le résidu)

Supposons qu'on ait sur un jeu de données de n observations, y_i la valeur de la i^{me} observation et Y_i sa valeur prédite par un modèle de régression.
On définit alors :

- le biais
  Le biais permet d'évaluer si les prédictions sont exactes ou non et si le modèle a tendance à surestimer ou sous-estimer les valeurs de la variable d'intérêt.

$$Biais = \frac{\sum_{i=1}^{n}(Y_i - y_i)}{n}$$

  En fait, si le biais est petit (près de 0), plus la prédiction est bonne.Il faut noter que cet indicateur ne tient pas compte de la variabilité des prévisions.En

effet, si les valeurs prédites sont à la fois très surestimées mais aussi très sous-estimées, le biais peut quand même être relativement faible.

- L'erreur moyenne absolue : MAE
  L'indicateur MAE permet d'avoir une idée de la qualité de prédiction.

$$MAE = \frac{\sum_{i=1}^{n} |Y_i - y_i|}{n}$$

  Elle quantifie l'erreur réalisée par le modèle. Plus elle est élevée, moins le modèle est performant.

- L'erreur quadratique moyenne : RMSE
  Le RMSE est l'écart-type des erreurs de prévision. Les résidus sont la mesure de l'écart entre les points de données et la ligne de régression. La métrique RMSE est la mesure de la répartition de ces résidus. En d'autres termes, elle indique la concentration des données autour de la courbe du meilleur ajustement.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - y_i)^2}{n}}$$

  Plus elle est élevée, moins le modèle est performant.

- L'erreur logarithmique quadratique moyenne : MSLE
  La MSLE est proche de la RMSE, mais les valeurs réelles et prédites sont remplacées par leurs logarithmes pour prendre en compte des variations exponentielles. Cela en fait une métrique adaptée lorsque les valeurs prédites varient sur une grande échelle.

$$MSLE = \frac{1}{n} \sum_{i=1}^{n} \left(ln(\frac{y_i + 1}{Y_i + 1}) \right)^2$$

  Elle quantifie l'erreur réalisée par le modèle. Plus elle est élevée, moins le modèle est performant.

- L'erreur moyenne médian
  Elle permet de calculer la médiane des résidus.

$$MDAE = median(|y_i - Y_i|)$$

  Elle est surtout utilisée lorsqu'il n'y a pas assez de grandes erreurs de prédiction du fait que les prédictions sont très grandes comparées aux vraies valeurs.

En effet, pour toutes ces métriques régressions liées aux mesures du résidu, il n'est pas possible de savoir si le modèle a tendance à sous ou surestimer les prédictions [6].

4.1.2 Le coefficient de détermination

Le coefficient de détermination noté R^2 permet d'évaluer la performance d'un modèle par rapport au niveau de variation présent dans le jeu de donnée. Il permet effectivement de savoir la tendance du modèle (sous ou surestimer les prédictions).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - Y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

avec $\bar{y}$: La moyenne des valeurs du jeu de donnée.

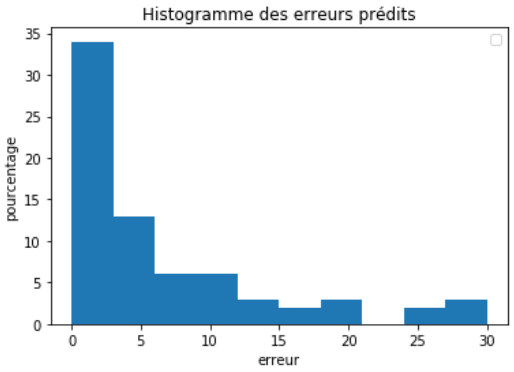
6. Sous ou Surestimer les prédictions, c'est savoir si les prédictions amplifient ou diminuent de manière générale comparée aux vraies valeurs.

4.2 L'analyse du modèle SIR+H

Dans cette section, nous avons appliqué les mesures antérieures aux prévisions du modèle SIR+H sous python.Nous avons ensuite évalué les 4 premières prévisions de ce modèle.

4.2.1 Les résultats de l'évaluation des prédictions

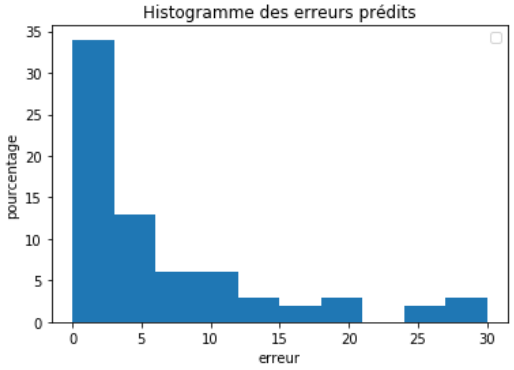
- Pour la 1^{er} prédiction, on a eu :
  $MAE = 6.8836, \quad RMSE = 10.3785, \quad MAX\_ERROR = 29.9642,$
  $MSLE = 1.7464, \quad MDAE = 3.5617, \quad R^2 = -0.7770.$



Source : Extrait après le lancement du programme d'évaluation des modèles de prédiction.

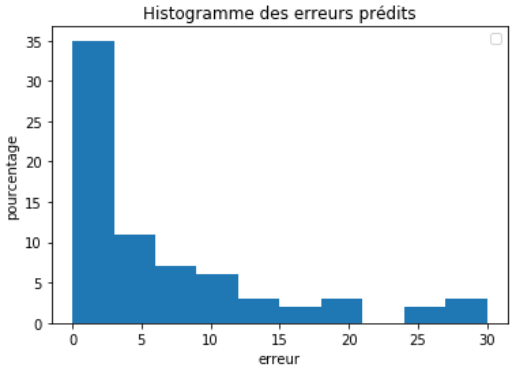
Il y a donc 35% des valeurs prédites qui ont une erreur proche de 0 ; c'est-à-dire 35% des prédictions avoisinent les données.

- Pour la 2^{e} prédiction, on a eu :
  $MAE = 6.8729, \quad RMSE = 10.3794, \quad MAX\_ERROR = 29.9665,$
  $MSLE = 1.7500, \quad MDAE = 3.6138, \quad R^2 = -0.7773.$



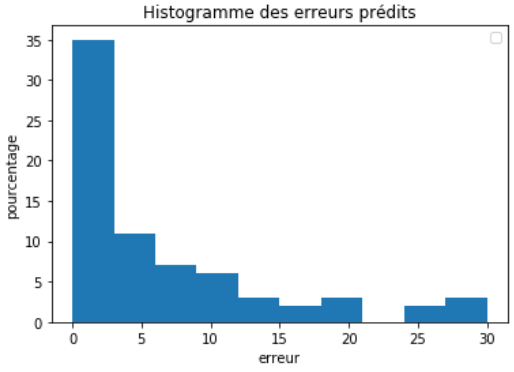
Il y a donc environ 35% des valeurs prédites qui ont une erreur proche de 0 ; C'est-à-dire que 35% des prédictions sont proches aussi des données.

- La 3^e prédiction, on a eu :
  $MAE = 6.8142, \quad RMSE = 10.3786, \quad MAX\_ERROR = 29.9688,$
  $MSLE = 1.7572, \quad MDAE = 3.4918, \quad R^2 = -0.7770.$



Il y a donc environ 35% des valeurs prédites qui ont une erreur proche de 0 ; C'est-à-dire que 35% des prédictions avoisinent les données.

- La 4^e prédiction, on a eu :
  $MAE = 6.8115, \quad RMSE = 10.3788, \quad MAX\_ERROR = 29.9688,$
  $MSLE = 1.7576, \quad MDAE = 3.5055, \quad R^2 = -0.7771.$



Il y a donc environ 35% des valeurs prédites qui ont une erreur proche de 0 ; C'est-à-dire que 35% des prédictions avoisinent aussi les données.

4.2.2 Interprétations

Pour les interprétations, nous nous servons des résultats de l'évaluation précédente du modèle SIR+H. Nous avons :

- Le MAE qui diminue avec chaque prédiction puis le modèle SIR+H devient progressivement performant, mais ce n'est pas vraiment significatif.

- Le $RMSE$ est presque pareil au niveau de chaque prédiction d'environ 10.37 ; alors la répartition des erreurs de prédiction est identique cela signifie a priori que les prédictions se dégradant généralement aux mêmes instants.

- Le MAX_ERROR est assez grand, il est d'environ 30 à chaque prédiction.

- Le $MSLE$ croit à chaque prédiction, mais reste inférieur à 2 ; alors la prédiction

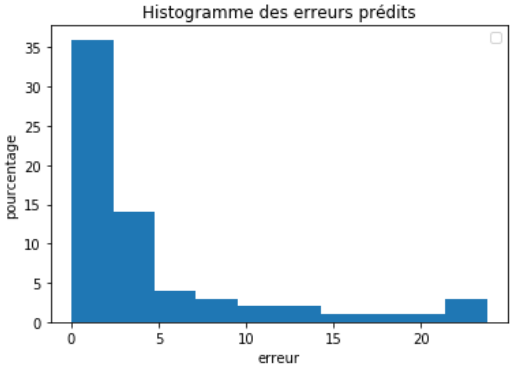
par le modèle SIR+H est de plus en plus imprécise.
- Le $MDAE$ tourne autour de 3.5, cela indique que 50% des prédictions ont une marge d'environ 3.5 ; ce qui n'est pas très idéal vu les données réelles.
- Au niveau de l'histogramme des erreurs, il y a environ 35% des prédictions qui avoisinent les données réelles.
- Le R^2 est presque identique et négatif ; cela montre que le modèle SIR+H a une forte tendance à sous-estimer les prédictions et que les erreurs de prédictions sont plus grandes que la dispersion des données réelles.

4.3 L'analyse du modèle GP

Dans cette partie, nous avons appliqué les précédentes métriques aux prédictions lu modèle GP sous python. Nous avons alors évalué les 4 premières prédictions du modèle.

4.3.1 Les résultats de l'évaluation des prédictions

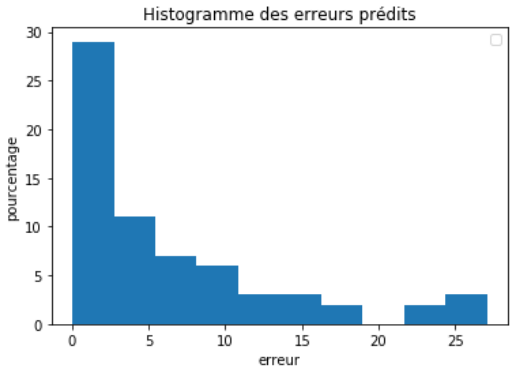
- Pour la 1^{er} prédiction, on a eu :
  $MAE = 1.6541, \quad RMSE = 1.6541, \quad MAX\_ERROR = 2.9719,$
  $MSLE = 0.9106, \quad MDAE = 1.6470, \quad R^2 = 0.1118.$



Source : Extrait après le lancement du programme d'évaluation des modèles de prédiction.

Il y a donc environ 36% des valeurs prédites qui ont une erreur proche de 0 ; c'est-à-dire que 36% des prédictions avoisinent les données.

- Pour la 2^e prédiction, on a eu :
  $MAE = 0.9310, \quad RMSE = 1.0404, \quad MAX\_ERROR = 1.3058,$
  $MSLE = 0.6687, \quad MDAE = 1.2299, \quad R^2 = -0.4247.$

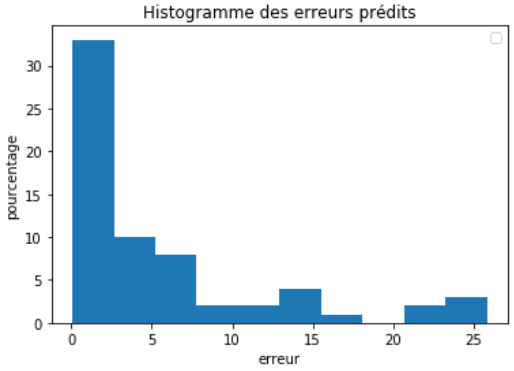


Il y a donc environ 30% des valeurs prédites qui ont une erreur proche de 0 ; c'est-à-dire que 30% des prédictions sont proches aussi des données.

- La 3^e prédiction, on a eu :

  $MAE = 0.8930, \quad RMSE = 1.0421, \quad MAX\_ERROR = 1.6656,$

  $MSLE = 0.6305, \quad MDAE = 0.7860, \quad R^2 = -0.7770.$

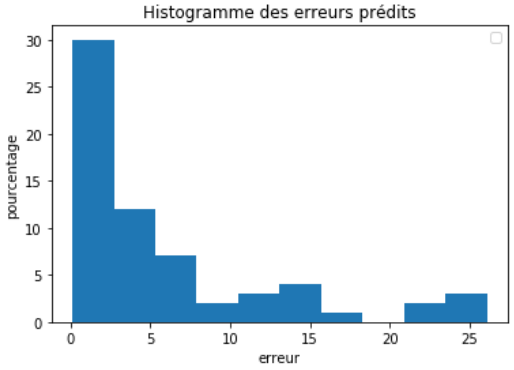


Il y a donc environ 35% des valeurs prédites qui ont une erreur proche de 0 ; c'est-à-dire que 30% des prédictions avoisinent aussi les données.

- La 4^e prédiction, on a eu :

  $MAE = 0.4881, \quad RMSE = 1.6346, \quad MAX\_ERROR = 2.1326,$

  $MSLE = 0.7627, \quad MDAE = 1.8351, \quad R^2 = -0.2890.$



Il y a donc environ 30% des valeurs prédites qui ont une erreur proche de 0 ; c'est-à-dire que 30% des prédictions avoisinent aussi les données.

4.3.2 Interprétations

Nos interprétations seront basées sur les résultats de l'évaluation précédente du modèle GP. Nous avons alors :
- Le MAE qui diminue à chaque prédiction alors le modèle GP devient performant au fur à mesure l'apprentissage des hyperparamètres.
- Le $RMSE$ est presque pareil au niveau de chaque prédiction d'environ 1.6 ; alors la répartition des erreurs de prédiction est identique cela signifie à priori que les prédictions se dégradant généralement aux mêmes instants.
- Le MAX_ERROR est variable, mais il reste inférieur à 3, cela signifie que la plus grande erreur effectuée par le GP est plus petite (≤ 3).
- Le $MSLE$ varie chaque prédiction, mais il reste inférieur à 1 ; alors la prédiction par le modèle GP est plus précise.
- Le $MDAE$ reste inférieur à 2, cela indique que 50% des prédictions ont une marge d'environ 2.
- Au niveau de l'histogramme des erreurs, à part la première où il y'a environ 36% des prédictions qui avoisinent les données réelles, toutes les autres prédictions ont environ 30% des prédictions qui avoisinent les données réelles. Cela signifie sur la première prédiction est la meilleure.
- Au niveau du R^2, à part pour la première prédiction où le modèle GP décrit environ 12% des variations des données réelles, les autres restent négatives inférieures à -1 ; cela montre qu'il a une tendance à sous-estimer les prédictions à partir de la deuxième, mais cet effet n'est pas vraiment significatif.

4.4 Bilan Comparatif

Au regard des interprétations des modèles, on remarque :
- Les métriques MAE, $RMSE$, MAX_ERROR, $MSLE$ et $MDAE$ du modèle de processus gaussien sont globalement plus petits que le modèle SIR+H avec une marge importante.En effet, cet aspect démontre clairement l'amélioration du processus GP sur le modèle SIR+H.
- Quant à la métrique R^2, on constate que les coefficients de détermination pour les prédictions avec GP sont aussi plus petits cependant il n'y a pas une grande marge de différence ; ce qui signifie que le modèle du GP décrit un peu plus mieux la dispersion des données réelles. De plus, comme signifié dans la section **??**, on observe qu'il y a environ 12% des premières prédictions du processus gaussien qui décrivent

parfaitement l'évolution réelle des données.
Les prédictions des modèles les moins erronées concernant la COVID-19 sont celles obtenues sur un à cinq jours. (cf. ref.[5]). Cela prouve effectivement que sur la première prédiction, les estimations du GP sont nettement mieux.
À l'égard de ces remarques, on peut conclut avec certitude, l'amélioration des prédictions du flux de patients qu'apporte le processus GP sur le modèle compartimental SIR+H.

Conclusion

Les travaux menés durant ce mémoire de recherche ont eu pour objectif de comprendre des modèles prédictifs développes par M. Bart Lamiroy et son équipe afin de les évaluer pour estimer leurs fiabilités vu qu'ils ont été mis en place pour "Sauver des vies humaines" durant la crise sanitaire lié au coronavirus en 2020, i.e. qu'ils permettront de prévoir le flux patient dans un hôpital à court terme afin que les dirigeants de l'hôpital universitaire de Nancy adoptent rapidement au mieux des mesures dans les unités de soins.

Le premier modèle développé est le modèle SIR+H (section.2.1), qui est un modèle similaire à un modèle épidémiologique très connu (le modèle SEIR) où certains critères et paramètres ont été modifié relativement à la situation réelle de cet hôpital durant cette crise. À ce niveau, nous avons pu constater que ce modèle va dépend de l'estimation des paramètres durant les phases de propagation du virus.
Le deuxième modèle développé est le modèle GP (section.2.2), qui un modèle d'apprentissage automatique appliqué pour traiter un problème de régression où le comportement des données suit une distribution gaussienne multivariée. Nous avons également constaté que ce modèle est tributaire de certains hyperparamètres liés aux caractéristiques des données réelles.

La réalisation de ses modèles (chap.3) s'est effectuée sur une base de données fournies par le service hospitalier donnant le nombre d'infectés à l'unité de réanimation sur la période du 06/01/2020 au 18/03/2020 et afin de tenir compte de la situation réelle de l'hôpital durant cette période, les paramètres initiaux du modèle SIR+H ont été calculé grâce à la base de données des indicateurs de suivi de l'épidémie en France.

Ainsi, après avoir appliqué les modèles à cette base de données, nous menant aux prédictions estimées au fur à mesure de la détermination des paramètres liés

aux modèles, on a pu apercevoir que les prédictions avec le SIR+H sont très vite dégénérées et n'avoisinent pas la dispersion de données. Quant aux modèles GP, on remarque qu'il y a une nette amélioration aux niveaux des prédictions puis celles-ci se dégradent progressivement en restant bien sûr à chaque fois préférable aux prédictions avec le modèle SIR+H.

De ce constat précédent, nous avons procédé à analyser séparément ces prédictions (chap.4) en nous servant de quelques métriques, et effectivement le modèle le plus fiable est celui du processus gaussien(GP).

Il va sans dire que, dans notre situation, il n'y a réellement aucune raison de comparer ces modèles étant donné que l'approche bayésienne bayésienne du modèle GP qui consiste à apprendre du modèle SIR+H pendant un certain temps afin de mieux décrire l'évolution future de la population infectée. On dira alors que ces modèles de prédictions sont complémentaires pour une bonne prise de décision qu'appliqueront les dirigeants de l'hôpital universitaire de Nancy afin de sauver le plus de vie humaine.

Bien qu'à l'échelle de l'évaluation des modèles, il existe inévitablement d'autres méthodes plus sophistiquées que nous n'avons pas pu utiliser pour mieux juger ces modèles. Cependant, tout au long de ce mémoire, j'ai développé des compétences très utiles dans le domaine des données et cela m'a permis de comprendre que l'un des objectifs des entreprises consiste à améliorer le rendement des services afin de les rendre plus optimaux.
De ce fait, ce mémoire ouvre de nouvelles perspectives puisque l'on sait que le duo de ces modèles prédictifs permet de prévoir l'évolution future du nombre d'infectés, mais il est encore peu significatif en termes de données, il serait intéressant :

- De trouver une meilleure méthode d'estimation des paramètres (beta, beta_post, patient0, dm_r, dm_incub) du modèle SIR+H prenant en compte la situation réelle au sein des services afin de disposer de prévisions qui dégénèrent moins rapidement.
- D'augmenter le temps d'évolution du modèle SIR+H, de sorte que le modèle GP en apprenne davantage sur le comportement de ces prédictions ultérieures afin d'avoir un modèle qui décrit mieux l'évolution des données réelles.

Annexes

Annexe du modèle SIR+H.

1 L'algorithme de l'estimation des paramétres du modele SIR+H

Dans cette partie, nous expliquerons le script defaults.py qui permet d'estimer les paramètres *(beta, beta_post, patient0, dm_r, dm_incub)* du modèle SIR+H puis les sauvegarde dans un fichier ".json" [7]. Ce script est composé de plusieurs fonctions qu'on détaillera :

- la classe **ModelParameters** comprend tous les paramètres globaux, elle est constituée de :

  - _init_(self) : la fonction où l'on définit la durée moyenne de rétablissement ($dm\_r : \gamma$), self._day0 : le début de la simulation, les dates des divers confinements, R_0 : le taux de reproduction initial et les taux de reproduction liés à chaque phase et pc : les pourcentages de transmission entre chaque compartiment (observés entre les unités de soins).

  - self._paramètres : la liste des tous les paramètres précédemment fixés avec en plus la taille de la population du modèle SIR+H (*population*), le nombre d'itérations de simulation à exécuter (*lim_time*), la durée moyenne d'incubation du virus ($dm\_incub : \alpha$), le nombre d'infectés au début de la simulation (*patient0*), le taux de transmission initial ($beta : R_0/dm\_r$).

  - self._data_chu_rea : la base de données du nombre d'infectés en soins de réanimation depuis le 01/01/2020.

  - self._rules : la table du taux de transmission (β) à chaque instant de la propagation du virus.

  - self._other : La liste des autres paramètres.

- Les fonctions qui renvoient les informations de la classe **ModelParameters**

7. Un fichier JSON est un fichier qui conserve des structures de données simples et des éléments au format JavaScript Object Notation (JSON).

séparément : **parameters (self)**, **data_chu_rea (self)**, **day0 (self)**, **other (self)**.

- La fonction **import_json (filename : str)** : permet de générer un fichier "json" contenant tous les paramètres (cf. fig.3.1).
- **get_default_params()** : elle retourne la liste de tous les paramètres, la table du taux de transmission, la base de données du nombre d'infectés, la liste des autres paramètres.

2 L'exemple simple de programmation du modéle SEIR en python

Dans cette partie, nous expliquerons le modèle SEIR avec un exemple simple en le programmant en python. Pour rappel, ce modèle est décrit par le système d'équations suivant :

$$\begin{cases} \frac{dS(t)}{dt} &= -\beta S(t)I(t) \\ \frac{dE(t)}{dt} &= \beta S(t)I(t) - \alpha E(t) \\ \frac{dI(t)}{dt} &= \alpha E(t) - \gamma I(t) \\ \frac{dR(t)}{dt} &= \gamma I(t) \end{cases}$$

Avec :
-**S**(t) : l'effectif de personnes sensibles le jour t.
-**E**(t) : l'effectif de personnes exposées le jour t.
-**I**(t) : l'effectif de personnes infectées le jour t.
-**R**(t) : l'effectif de personnes guéries le jour t.
-α : le taux d'incubation du virus.
-β : le taux de transmission par jour.
-γ : le taux de rétablissement guérissent par jour.

On suppose comme effectif de départ : $S = 3000, \quad E = 100, \quad I = 50, \quad R = 500,$ puis on cherche l'évolution des compartiments sur 100 jours. Pour la programmation, on utilise plus particulièrement la package "scipy.integrate" qui permet de résoudre les équations différentielles ordinaires sur Python.

Puis nous choisissons les paramètres du modèles(α,β,γ). On trouve ainsi l'évolution des 4 populations de ce modèle.

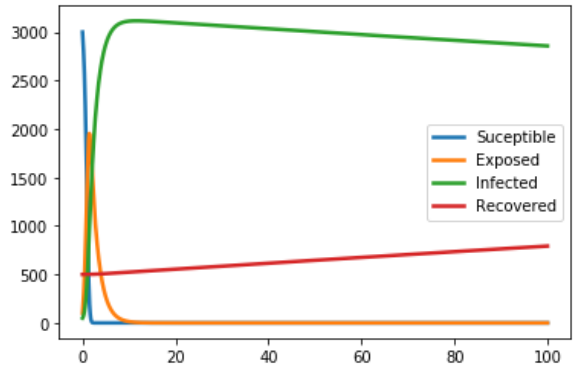


FIGURE 4.1 – La proportion des individus durant une épidémie par le modèle SEIR avec alpha=0.6, beta=0.005, gamma=0.001

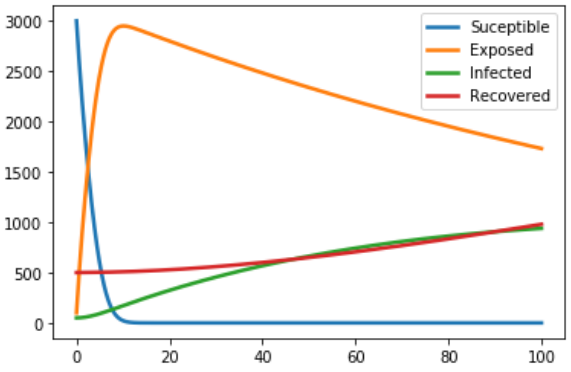


FIGURE 4.2 – La proportion des individus durant une épidémie par le modèle SEIR avec alpha=0.006, beta=0.005, gamma=0.008

On observe que l'effectif des infectés et des exposés évoluent différemment ; on a un pic des infectés qui est de 3000 sur la figure 4.1 et un pic des exposés qui est de 3000 sur la figure 4.2, cela s'explique par le fait que les taux d'incubation dans les deux situations ont une grande marge. Quant à l'effectif des rétablies, elle n'a pas tellement varié car le taux de transmission du virus est très faible et n'a pas changé.

```python
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt

# Description des 3 equations differentielles
 #S,E,I,R = U
def dU_dt(U,t):
    return [-beta*U[0]*U[2],
            beta*U[0]*U[2]-alpha*U[1],
            alpha*U[1]-gamma*U[2], gamma*U[2]]

#les conditions initiaux
U0=[3000, 100, 50, 500]

#les parametres  du modele
alpha, beta, gamma =0.006, 0.005, 0.008  #incubation transmission guerison


#Evolution sur 100 jours
ts = arange(0, 100, 0.01)

# Resolution du systeme EDO
Us= odeint(dU_dt, U0, ts)
S,E,I,R= Us[:,0], Us[:,1], Us[:,2], Us[:,3]

#Graphe
fig=plt.subplots(figsize=(6,4))
plt.plot(ts,S, linewidth=2.5, label='Suceptible')
plt.plot(ts,E, linewidth=2.5, label='Exposed')
plt.plot(ts,I, linewidth=2.5, label='Infected')
plt.plot(ts,R, linewidth=2.5, label='Recovered')
plt.legend(fontsize=10)
plt.show()
```

Source : Le programme développé d'un modèle SEIR en python.

FIGURE 4.3 – L'exemple de Script en python du modèle SEIR en Python

Annexe sur le processus gaussien

1 L'apprentissage des hyperparamètres

Ils existent plusieurs méthodes pour l'apprentissage des hyperparamètres qui sont : l'optimisation basée sur le gradient, la méthode de Monte Carlo par la chaîne de Markov (MCMC),...

Pour la suite, j'expliquerai la méthode basée sur l'optimisation de gradient (cf. ref.[11]).

D'abord, on suppose le vecteur des hyperparamètres de l'ensemble des données d'apprentissage $\theta = [\sigma, l, \sigma_y^2]$. Comme le noyau k mesure la similarité et la variance du bruit, on a : $p(f(X)|X, \theta) \sim N(0, K(X, X))$ et $p(y|f(X), \theta) \sim N(f(X), \sigma_y^2 I)$.

L'idée sera de sélectionner ceux qui maximisent la probabilité logarithmique de y après l'idée sera de sélectionner ceux qui maximisent la probabilité logarithmique de y après avoir intégré les fonctions possibles ($f(x) \sim GP(0, K(x, x'))$).

$$logp(y|X, \theta) = log \int p(y|X, \theta)p(f(X)|X, \theta)df(X)$$

$$logp(y|X, \theta) = -\frac{1}{2}y^T(K(X, X) + \sigma_y^2 I)^{-1}y - \frac{1}{2}log[K(X, X) + \sigma_y^2 I] - \frac{n}{2}log(2\pi)$$

Ensuite, les dérivées partielles de cette équation trouvée (par rapport aux hyperparamètres θ) sont utilisées dans l'optimisation basée sur le gradient pour apprendre les hyperparamètres qui maximisent la probabilité marginale de l'équation. La vraisemblance marginale de l'équation. En outre, il convient de mentionner qu'il n'y a aucune garantie d'atteindre un optimum global puisque ce problème d'optimisation n'est pas convexe. Dans la pratique, cependant, cette approche d'optimisation basée sur le gradient a tendance à bien fonctionner (cf. ref.[4].[8]).

2 L'exemple simple du processus gaussien en 1D en python

On se donne ici une table de donnée avec 2 entrées (X est un vecteur et $Y = X.\sin X$), où l'on déterminera la fonction de régression de cette de donnée par l'algorithme du GP et puis on effectuera des prédictions avec un autre vecteur afin de voir si elles concordent avec les vraies valeurs.
Nous avons alors $X = (1, 0, 5, -1, 7, -5, 18, 3, -13, 9)$, on aura alors :

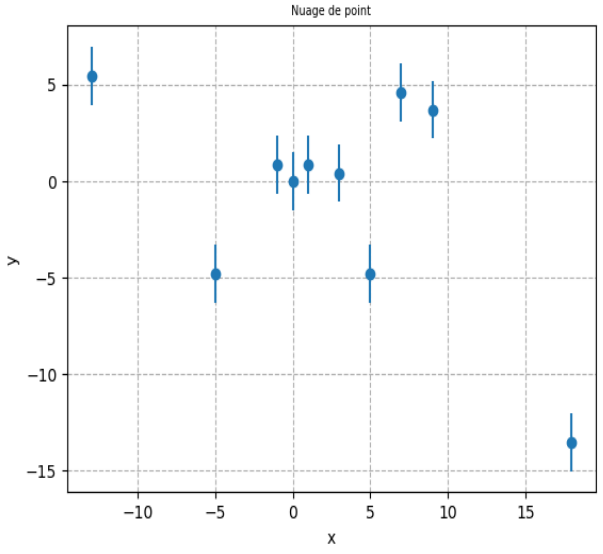
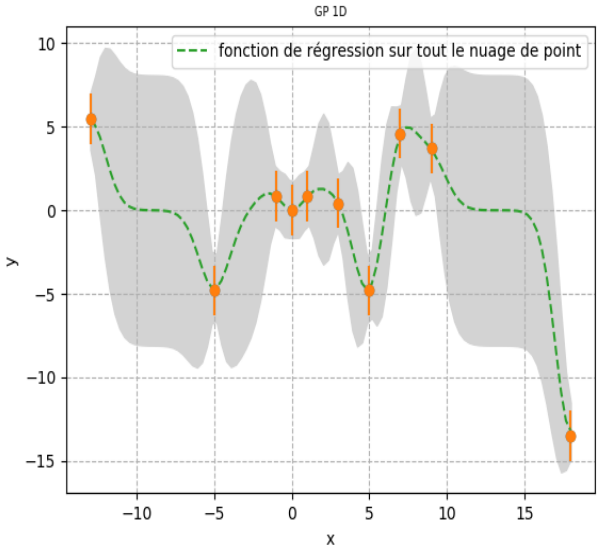


FIGURE 4.4 – Nuage de points sur nos données d'apprentissage

On suppose le vecteur des hyperparamètres de l'ensemble des données d'apprentissage $\theta : [\sigma = 10, \quad l = 1, \quad \sigma_Y^2 = 1,5]$.
Pour le calcul de la fonction de covariance $K(X, X')$, on utilise l'exponentielle au carré (cf. section.2.2.1) et on trace la fonction f de régression au fur à mesure en ne tenant que des données antérieures car $Y = f(X) + \epsilon \quad avec \quad \epsilon \sim N(0, \sigma_y^2)$.
Puis on ajoute sur le graphe une région de confiance avec une marge d'erreur ($[m - \sigma, m + \sigma]$,le bruit d'observation) .
Pour tester cette fonction de régression trouvée, on choisit $X\_test = 4.0$. La fonction GP nous renvoie comme valeur prédite $y\_predict = -3.027101e - 10$ qui est une valeur très différente de la donnée réelle qui est égale $Y = 4.0 * \sin(4.0) = -3.02$.
Cela n'est pas étonnant vu que l'algorithme prédictif du GP dépend fortement du choix de hyperparamètres, c'est donc pour cela qu'il est important de faire l'apprentissage des hyperparamètres (cf. section.2.2.3) afin d'avoir un meilleur modèle



Source : Extrait après le lancement du programme du processus gaussien en 1D.

FIGURE 4.5 – la fonction de régression de GP sur tout le nuage de point

prédictif. Par exemple, lorsqu'on on choisit $\theta : [\sigma = 4,0009, \quad l = 7,05, \quad \sigma_Y = 5]$, on trouve $y\_predict = -3.3032360$ qui bien plus proche de la donnée réelle de X_test.

```python
from numpy.linalg import inv
from scipy import misc
import matplotlib.pyplot as plt
import numpy as np
"""..................La table de données....................."""
X = np.array([1., 0., 5., -1., 7.,-5.,18.,3.,-13., 9.])
Y = X * np.sin(X)
X = X[:,np.newaxis]  #transformer X en colonne
"""..................Nuage de points....................."""
sigma_n = 1.5 #sigma_n=5 #\sigma_{Y}
plt.grid(True,linestyle='--')
plt.errorbar(X, Y, yerr=sigma_n, fmt='o')
plt.title('GP 1D', fontsize=7)
plt.xlabel('x')
plt.ylabel('y')
"""..................Calculer la matrice de covariance K....................."""
sigma_f = 10.0  #sigma_f=4.0009 #\sigma
l=7.05#l = 1.0
X_dim1 = X.shape[0] #dim de X
D = np.zeros((X_dim1,X_dim1)) #matrice nulle 6x6
K = np.zeros((X_dim1,X_dim1)) #matrice nulle 6x6
D = X - X.T # X-X'
K = sigma_f**2*np.exp((-D*D)/(2.0*l**2)) #matrice de covariance
np.fill_diagonal(K, K.diagonal() +sigma_n**2 )
"""..................Faire une fonction prediction GP sur tout le nuage....................."""
X_new = np.linspace(-13,25,100)
Y_predict = []
Y_VAR_predict = []
plt.errorbar(X, Y, yerr=sigma_n, fmt='o') #Vrai graphe
"""..................graphe de prediction....................."""
for x_new in X_new:
    D_new = np.zeros((X_dim1))
    K_new = np.zeros((X_dim1))
    D_new = X - x_new
    K_new = sigma_f**2*np.exp((-D_new*D_new)/(2.0*l**2))
    m1 = np.dot(K_new[:,0],K_inv)
    y_predict = np.dot(m1,Y)
    Y_predict.append(y_predict)
    y_var_predict = K[0,0] - K_new[:,0].dot(K_inv.dot(np.transpose(K_new[:,0])))
    Y_VAR_predict.append(y_var_predict)
plt.plot(X_new,Y_predict, '--',label='fonction de régression sur tout le nuage de point')
plt.legend()
"""..................Tracer la région de confiance....................."""
plt.fill_between(X_new, [i-0.8*np.sqrt(Y_VAR_predict[idx]) for idx,i in enumerate(Y_predict)],
[i+0.8*np.sqrt(Y_VAR_predict[idx]) for idx,i in enumerate(Y_predict)],color='#D3D3D3')
```

Source : Le programme développé du processus gaussien en 1D.

FIGURE 4.6 – Exemple de Script en python du processus gaussien en 1D

Annexe sur l'évaluation des modèles

1 La méthode de validation croisée (cross validation)

Elle est l'une des techniques les plus utilisées pour tester l'efficacité d'un modèle de Machine Learning comme le cas du processus Gaussien. Elle est aussi une procédure de rééchantillonnage, i.e. qu'elle permet d'évaluer un modèle même avec des données limitées.
Les approches de cette méthode qui sont souvent utilisées sont :

- la technique "train-test split" qui consiste à diviser de manière aléatoire l'ensemble d'observations en deux parties : un ensemble d'apprentissage (A) et un ensemble de validation (V). Le modèle est construit à partir des données d'apprentissage, et il est ensuite utilisé pour prédire la variable réponse des données de l'ensemble de validation.
- La technique "K-Folds" qui consiste à diviser de manière aléatoire le tableau des données en K groupes ("folds") de même effectif (avec K entre 5 et 10). Notons ces groupes G1,..., GK. Le premier groupe est considéré comme un ensemble de validation, les $K - 1$ autres groupes sont utilisés pour ajuster le modèle.

Malheureusement, ces deux techniques de la méthode de validation croisée n'ont pas pu être utilisées du fait des démarches du modèle SIR+H et GP. Comme raison, il y a le fait que les données de test sont choisies aléatoirement et le modèle prédictif est implanté directement sur le logiciel (comme sur Python ou Rstudio) afin d'évaluer les données de test.

2 L'algorithme de l'évaluation des modèles en python

Pour l'évaluation des modèles, nous nous sommes servis du package sklearn.metrics qui contient assez de métriques déjà défini sur python. Cela a été très bénéfique pour notre travail, nous avons aussi tracé l'histogramme des erreurs de prédiction en pourcentage afin de connaître la proportion des prédictions qui avoisinent les données.

```
## Importation de la data
data= pd.read_csv('predictgp1.csv')
predict1=data.iloc[:,-1].values #predictions
val=data.iloc[:,-2].values # valeur
t1=data.iloc[:,:-3].values  #jour
## Evaluation du modele
   # Calcul du MAE
print('MAE:', mean_absolute_error(val,predict1))
          #lerreur moyenne absolue est ..... ce qui est trés petit
   # Calcul du RMSE
print('RMSE:', sqrt(mean_squared_error(val,predict1)))
          #La repartition des erreurs est ......
   # Calcul du MAX_error
print('Max_error:', max_error(val,predict1))
          # La plus grande erreur de prédiction est ......
   # Calcul du MSLE
print('MSLE:', sqrt(mean_squared_log_error(val,predict1)))
          # MSLE= ..... ce qui n'est pas si mal comme prediction
   # calcul MDAE(Erreur médian)
print('median abs err:', median_absolute_error(val,predict1))
          #la mediane des erreurs de prediction est .......
   # L'histogramme des erreurs
data= pd.read_csv('gp1.csv')
predict1=data.iloc[:,-1].values #predictions
val=data.iloc[:,-2].values # valeur
err_hist1= abs(val-predict1)
plt.hist(err_hist1)
plt.title('Histogramme des erreurs prédits')
plt.xlabel('erreur')
plt.ylabel('pourcentage')
plt.legend()
        #environ .... des valeurs predites ont une erreur proche de 0
   #Calcul de R^2
r2=r2_score(val,predict1)
print(r2)
       # R2=......
```

Source : Le programme développé du processus gaussien en 1D.

FIGURE 4.7 – Script d'évaluation des modèles prédictifs

Bibliographie

[1] Samuel Alizon, Bastien Reyné, and Christian Selinger. *Modélisation de l'épidémie de COVID-19 : modèle SEAIR*. PhD thesis, Centre national de la recherche scientifique (CNRS) ; Institut de Recherche . . ., 2020.

[2] Derdei Bichara. *Étude de modèles épidémiologiques : Stabilité, observation et estimation de paramètres*. PhD thesis, Université de Lorraine, 2013.

[3] Alexis Boukouvalas, Remi Barillec, and Dan Cornford. Gaussian process quantile regression using expectation propagation. *arXiv preprint arXiv :1206.6391*, 2012.

[4] Carl Edward. Rasmussen et christopher ki williams. processus gaussiens pour l'apprentissage automatique. *MIT Press*, 211 :212.

[5] Patrick Giraudoux. *Modèle de prévision statistique COVID-19 basé sur 10 jours d'observation*. PhD thesis, Université de Franche-Comté Besançon ; UMR 6249 Chrono-environnement, 2020.

[6] Ali Hebbal. *Processus gaussiens profonds pour l'analyse et l'optimisation de systèmes complexes-Application à la conception de systèmes aérospatiaux*. PhD thesis.

[7] Ananth Ranganathan. The levenberg-marquardt algorithm. *Tutoral on LM algorithm*, 11(1) :101–110, 2004.

[8] Jakub M Tomczak, Jerzy Swiatek, and Krzysztof Latawiec. Gaussian process regression as a predictive model for quality-of-service in web service systems. *arXiv preprint arXiv :1207.6910*, 2012.

[9] Suzanne Touzeau. Modèles épidémiologiques (r0). *UR341 Mathématiques et Informatique Appliquées*, 2010.

[10] Franck Varenne. Préface à" la diffusion de la covid-19-que peuvent les modèles ?", 2020.

[11] Ya Li Wang. Interactions between gaussian processes and bayesian estimation. 2014.

[12] Howard Howie Weiss. The sir model and the foundations of public health. *Materials matematics*, pages 0001–17, 2013.

[10]

Les explorations Internet

- Son github M.Bart Lamiroy sur le package FLOWSIM.[en ligne].(page consultée le 10/05/2022).
  <https://github.com/lamiroy/flowsim>
- Quand prédire ne laisse rien hasard : comment l'IA proule l'analyse prédictive.[en ligne].(page consultée le 10/05/2022).
  <https://www.lesechos.fr/partenaires/orange-business-services/quand-predire-
  ~:text=Les%20mod%C3%A8les%20pr%C3%A9dictifs%20par%20l,produits%20potentielle
  20pertinents%20aux%20consommateurs.>
- la dataset sur les indicateurs de suivi de l'épidémie en France.[en ligne].(page consultée le 10/05/2022).
  <https://www.data.gouv.fr/fr/datasets/indicateurs-de-suivi-de-lepidemie-de-

- Comprendre le processus Gauss, la voie socratique.[en ligne](page consultée le 10/05/2022).
  <https://ichi.pro/fr/comprendre-le-processus-gaussien-la-voie-socratique-12
- Qu'est ce qu'un modèle prédictif ?[en ligne].(page consultée le 10/05/2022).
  <https://kobia.fr/quest-ce-quun-modele-predictif/>
- Qu'est-ce que la modélisation prédictive ?.[en ligne].(page consultée le 10/05/2022).
  <https://ia-data-analytics.fr/modelisation-predictive/>
- Comment valider un modele de prediction ?[en ligne].(page consultée le 10/05/2022).
  <https://www.aspexit.com/comment-valider-un-modele-de-prediction/>
- Evaluez les performances d'un modele de machine learning[en ligne].(page consultée le 10/05/2022)
  <https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-
  4308276-evaluez-un-algorithme-de-regression>
- Guide des métriques de régression[en ligne].(page consultée le 10/05/2022)
  <https://kobia.fr/regression-metrics-quelle-metrique-choisir/>

- La Modélisation d'une épidémie(partie 2). [en ligne].(page consultée le 10/05/2022). <http://images.math.cnrs.fr/Modelisation-d-une-epidemie-partie-2.html>
- La Modélisation d'une épidémie (partie 1).[en ligne].(page consultée le 10/05/2022). <http://images.math.cnrs.fr/Modelisation-d-une-epidemie-partie-1.html>
- Les Modèles compartimentaux en épidémiologie [en ligne].(page consultée le 10/05/2022). <https://fr.wikipedia.org/wiki/Modèles_compartimentaux_en_épidémiologie>
- Les confinements liés à la pandémie de Covid 19 en France.[en ligne].(page consultée le 10/05/2022). <https://fr.wikipedia.org/wiki/Confinements_liés_à_la_pandémie_de_Covid-19_en_France>
- Comment utiliser les processus Gaussien avec pour faire une régression ou une classification (en "machine learning) avec python 3.[en ligne].(page consultée le 10/05/2022). <https://moonbooks.org/Articles/Introduction-aux-processus-Gaussien-avec-py
- Les MÉTRIQUES de RÉGRESSIONS en DATA SCIENCE.[en ligne].(page consultée le 10/05/2022). https://www.youtube.com/watch?v=_TE9fDgtOaE&t=1s