Université Cheikh Anta Diop

Faculté Des Sciences Et Techniques Département de Mathématiques-Informatique

Mémoire présenté pour l'obtention du diplôme de

Master en Informatique

Spécialité : **Business Intelligence**

Par

EL HADJI ABDOULAYE DIANKHA

Sur le sujet

Analyse et prédiction de la survie d'un patient sous traitement de la COVID19 : CAS de l'Hôpital Abass Ndao.

Soutenu le 08 aout 2023 devant le jury composé comme suit :

Président : Pr. Karim KONATÉ, Professeur Titulaire CAMES, UCAD

Examinateur :
Dr. Mamadou THIONGANE, Maître Assistant CAMES, UCAD
Dr. Mouhamed Ould DEYE, Maître-Assistant-CAMES, UCAD

Directeur de Mémoire : Pr. Samba NDIAYE, Professeur Titulaire CAMES, UCAD

Encadrant : Dr. Djamal Abdoul Nasser SECK, Maître-Assistant-CAMES, UCAD

Année universitaire : 2021-2022

DEDICACE

Ce travail est spécialement dédié à mes parents, pour leurs conseils et prières qui m'ont toujours accompagné, mais aussi et surtout pour cette brillante éducation qu'ils ne cessent de me donner depuis ma tendre enfance. Je leur dédie ce travail aussi pour l'amour qu'ils portent à mon égard, les sacrifices et surtout le lourd investissement qu'ils ont fait uniquement pour la bonne conduite de mon cursus scolaire.Je dédie aussi ce travail à mes frères et sœurs.

Qu'ils trouvent ici, le témoignage de ma très grande affection.

REMERCIEMENTS

Je rends grâce à Allah le tout puissant de m'avoir façonné avec cette belle forme humaine, de m'avoir ouvert l'esprit pour apprendre afin de servir et surtout de m'avoir gratifié des parents modèles qui ont fortement contribué à ma formation.

Je remercie le président du jury d'avoir accepté de présider le jury de soutenance de mon mémoire, ainsi que tous les autres membres du jury.

Je tiens à remercier toutes les personnes qui, de près ou de loin, m'ont soutenu dans la réalisation de ce travail. Particulièrement j'exprime ma plus vive reconnaissance à mon encadreur Dr Djamal SECK pour ses conseils, ses remarques, mais aussi et surtout pour sa grande disponibilité à mon égard tout au long de ce travail. Vraiment un grand merci à vous Dr...

Je suis redevable aux membres des corps professoral et administratif du master en informatique spécialité business intelligence de l'Université Cheikh Anta DIOP de Dakar. Je les remercie infiniment pour leur courage et leur dévouement au travail.

Mes sincères remerciements vont à l'endroit de mes camarades de classe pour ces beaux moments qu'on a eu à passer ensemble. Mes sincères remerciements vont aussi à l'endroit de mes frères et sœurs.

Enfin, je ne saurai terminer sans pour autant remercier mes collègues, toute l'équipe de la SOBOA Sénégal ; sans votre compréhension ce travail aurait pu être plus difficile.

Résumé

L'objectif de ce travail de mémoire est de faire, dans un premier temps, une analyse de la survie de patients sous traitement de la COVID-19 à l'hôpital Abass Ndao de Dakar et, dans un deuxième temps, de faire la prédiction de la survie des patients en fonction d'un ensemble d'attributs.

Pour l'analyse de survie nous avons utilisé l'estimateur de survie de Kaplan Meier qui est une méthode non paramétrique souvent préférable pour estimer la fonction de survie appelée la probabilité de ne pas échouer ou de survivre jusqu'à un certain temps. Avec cet estimateur nous avons pu analyser la probabilité de la survie des patients sur divers groupes tels que le sexe et la tranche d'âge afin de savoir la différence significative dans le taux de survie. Cette solution permet à un médecin ou un décideur de l'hôpital d'avoir une vue globale et de suivre tous les patients jusqu'à la survenue de l'événement considéré.

Pour la prédiction de la survie nous avons entraîné plusieurs modèles sur notre jeu de données avec des algorithmes d'apprentissage supervisé comme la Régression Logistique, les Machines à Vecteurs de Support (SVM), le Random Forest et le Gradient Boosting. Nous avons choisi le meilleur de ces modèles selon des critères de performance tels que l'exactitude (Accuracy), la précision, le rappel (Recall) et le F1 score et nous l'avons implémenté sous forme d'application web avec une interface intuitive pour faciliter son utilisation

Mots-clés : Analyse de survie, Prédiction de survie, estimateur de survie de Kaplan Meier, Machine learning, apprentissage supervisé, modèle de prédiction.

abstract

The aim of this dissertation is, firstly, to analyze the survival of patients undergoing COVID-19 treatment at Abass Ndao Hospital in Dakar and, secondly, to predict patient survival as a function of a set of attributes.

For the survival analysis we used the Kaplan Meier survival estimator, which is a non-parametric method often preferred for estimating the survival function called the probability of not failing or surviving to a certain time. With this estimator we were able to analyze the probability of patient survival over various groups such as gender and age range to find out the significant difference in survival rate. This solution enables a doctor or hospital decision-maker to take a global view and follow all patients until the event in question occurs.

For survival prediction we trained several models on our dataset with supervised learning algorithms such as Logistic Regression, Support Vector Machines (SVM), Random Forest and Gradient Boosting. We selected the best of these models according to performance criteria such as accuracy (Accuracy), precision, recall (Recall) and F1 score, and implemented it as a web application with an intuitive interface for ease of use.

Keywords : survival analysis, survival prediction, Kaplan Meier survival estimator, machine learning, supervised learning, prediction model.

Table des matières

Table des figures

INTODUCTION GENERALE

0.1 Contexte

Le Covid-19 découvert à Wuhan en Chine en décembre 2019 a été officiellement déclaré par l'Organisation Mondiale de la Santé comme urgence de santé publique de portée internationale en janvier 2020.

La grande infectiosité de cette maladie a entraîné diverses mesures afin de limiter sa propagation : la distanciation sociale, le port d'un masque facial dans les lieux publics, le lavage régulier des mains.

Ce virus a eu, entre autres, des effets conséquents sur notre système de soins de santé. Parmi ces effets, il faut noter une réorganisation des soins de première ligne pour enrayer le risque de transmission du virus. Cette pandémie d'ordre mondial a bouleversé les systèmes de santé des pays prenons l'exemple du Sénégal ou dans nos structures de santé beaucoup de décès sont dû à une prise en charge tardive du patient.

Par conséquent, plusieurs équipes de recherche à travers le monde se sont lancées dans la fabrication de vaccins et d'approches à long terme axée sur la technologie afin de lutter contre cette pandémie.

0.2 Problématique

À l'hôpital ABASS NDAO, les décideurs utilisent Excel avec ses diagrammes croisés dynamique pour analyser leurs informations. Cette pratique a eu, entre autres, des effets négatifs sur leur prise de décisions.

Parmi ces effets, il faut noter :

- Difficulté à avoir une vue globale et à suivre tous les patients jusqu'à la survenue de l'événement considéré.
- Difficulté à prédire l'État futur d'un patient déjà consulté.

0.3 Objectifs

L'objectif de ce mémoire est de développer deux approches à long terme axées sur la technologie permettant à un décideur (médecin) de prédire la survie de chaque patient et d'analyser la probabilité de survie des patients sur divers groupes afin de savoir la différence significative dans le taux de survie de manière rapide et efficace.

0.4 Solution proposée

Aujourd'hui, avec le développement rapide de la technologie, il existe des techniques permettant d'agir plus tôt avec un patient sous traitement de Covid-19. Il s'agit par exemple de l'apprentissage automatique encore appelé Machine Learning qui a pour objectif principal d'apprendre le comportement d'une situation à partir d'étude faite sur des données à grande échelle afin de prédire le comportement d'autres nouvelles données de mêmes propriétés et les domaines particuliers des statistiques appelés analyse de survie qui a pour objectif principal

d'analyser la probabilité de survie des patients sur divers groupes afin de savoir la différence significative dans le taux de survie.

- Pour réaliser l'analyse de survie la solution proposée est d'utiliser l'estimateur Kaplan Meier.
- Pour réaliser la prédiction de survie, la solution proposée est d'entraîner plusieurs modèles avec des algorithmes d'apprentissage supervisé sur nos données et choisir le meilleur selon des critères de performance pour faciliter la tâche au décideur afin de lui permettre de prédire la survie d'un patient.

0.5 Plan du mémoire

Ainsi pour la réalisation de ce travail, nous avons réparti notre étude en 5 chapitres :

- **- Chapitre 1 :** Généralités sur l'analyse de survie. Dans ce chapitre, nous parlerons de l'analyse de survie, des différentes fonctions de survie et de l'estimateur kaplan meier.
- **- Chapitre 2 :** Création de la courbe de survie et générations des tables d'évènement afin de faire l'analyse de la survie des patients sur divers groupes afin de savoir la différence significative dans le taux de survie.
- **- Chapitre 3 :** Généralités sur l'apprentissage automatique. Nous parlerons du Machine Learning, des différentes familles d'apprentissage existantes et des méthodes de classification.
- **- Chapitre 3 :** : Construction du modèle de prédiction de la survie d'un patient. Dans ce chapitre, il s'agira d'évaluer certaines des méthodes de classification, d'en retenir celle qui a le meilleur score et de construire notre model.
- **- Chapitre 4 :** : Construction du modèle de prédiction de la survie d'un patient. Dans ce chapitre, il s'agira d'évaluer certaines des méthodes de classification, d'en retenir celle qui a le meilleur score et de construire notre model.
- **- Chapitre 5 :** : Création de l'application web qui intègre le modèle. Dans ce dernier chapitre, il s'agira de développer une interface web permettant aux utilisateurs désirant se consulter de remplir leurs données et de voir le résultat.

0. **Analyse et prédiction de la survie d'un patient sous traitement de la COVID19 : CAS de l'Hôpital Abass Ndao.**

Chapitre 1

Généralités sur l'analyse de survie

1.1 INTRODUCTION À L'ANALYSE DE LA SURVIE

Les données de survie se distinguent par une discipline statistique particulière. Ce premier chapitre essaie d'une part, de décrire ce que sont les données de survie et les données censurées, et d'autre part, de donner un aperçu des modèles de survie paramétriques et semi paramétriques, et des méthodes non paramétriques.

1.2 Données de survie et censure

1.2.1 Données de survie

Les données de survie représentent le temps écoulé entre le début d'une observation et l'arrivée d'un événement. Le cas d'événement le plus simple est le décès ; cependant, le terme « donnée de survie » couvre d'autres événements, comme l'apparition d'une maladie ou d'une épidémie. Dans l'industrie, il peut s'agir du bris d'une machine, ou en économie, du temps écoulé pour qu'une personne accepte un travail. Dans plusieurs cas, l'événement est la transition d'un état à un autre. Par exemple, le décès est la transition de l'état « vivant » vers l'état « mort ». L'apparition d'une maladie est la transition de l'état « en santé » vers l'état « malade ». La figure (1.1) illustre ces deux exemples. Selon le contexte, les termes décès, événement, échec ou transition peuvent être utilisés pour désigner l'événement constaté, et plus précisément ce qui se passe au temps de la réponse. Dans certain cas, l'aspect intéressant est la transition correspondant à l'incidence d'une maladie, et dans d'autres cas, l'aspect intéressant est l'état de la disparition d'une maladie (Hougaard, 1999).

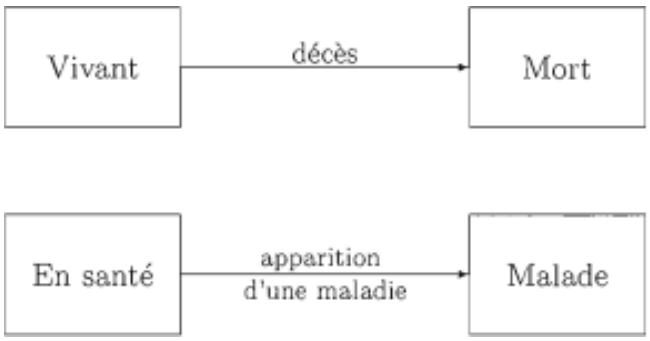


FIGURE 1.1 – Modèles de temps de survie

1.2.2 Données censurées

Une caractéristique importante de l'analyse de la survie est la présence des données censurées. Cette caractéristique, source de difficulté, a nécessité le développement de techniques alternatives à l'inférence usuelle. Les données censurées sont des observations pour lesquelles la valeur exacte d'un événement n'est pas toujours connue. Cependant, nous disposons tout de même d'une information partielle permettant de fixer une borne inférieure (censure à droite) ou une borne supérieure (censure à gauche). Les raisons de cette censure peuvent être le fait que le patient soit toujours vivant ou non malade à la fin de l'étude, ou qu'il se soit retiré de l'étude pour des raisons personnelles (immigration, mutation professionnelle). Il existe trois catégories de censure qu'on nomme censure à droite, censure à gauche et censure par intervalle (lorsqu'on connait la borne supérieure et la borne inférieure d'un événement).[1]

À l'intérieur de ces trois catégories, il existe différents types de censure :

- **- Censure de type I :** si le temps de censure est fixé par le chercheur comme étant la fin de l'étude. 2.
- **- Censure de type II :** se caractérise par le fait que l'étude cesse aussitôt qu'a eu lieu un nombre d'événements prédéterminé par l'expérimentateur.
- **- Censure aléatoire :** : slorsque le moment de censure n'est plus sous le contrôle du chercheur et/ou que le temps d'entrée varie aléatoirement. (Klein et Moeschberger, 2003). La figure 1.2 illustre les situations de censure de type l et de censure aléatoire. Un rond vide indique une censure et une croix indique un événement. Dans le premier graphique (celui de gauche), les censures de types l sont déterminées par la fin de l'étude, tandis que les censures du deuxième graphique varient aléatoirement et peuvent surgir avant la fin de l'étude (les censures A et E dans le graphique de droite).[1]

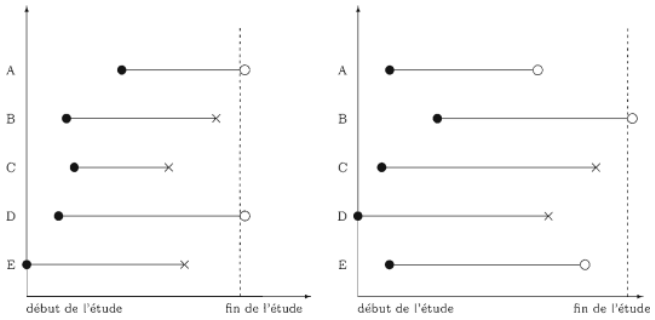


FIGURE 1.2 – Illustration de censure de type l (à gauche) et de censure aléatoire (à droite)

1.3 La fonction de risque et la fonction de survie.

Le développement de la méthodologie de l'analyse de la survie a connu un énorme progrès, et les premiers efforts ont été concentrés de façon prédominante sur l'estimation de la fonction de survie S(t). Dans l'analyse des données de survie (censurées) provenant d'études médicales, la fonction de risque est très utile. Elle contient de l'information sur le changement de risque

comme fonction de temps. Cette quantité fondamentale de l'analyse de la survie est définie par :

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} \tag{1.1}$$

Où T est une variable aléatoire non négative représentant le temps de survie d'un individu. Ce conditionnement successif fait en sorte que la fonction de risque est le concept le plus pertinent, car il décrit la probabilité qu'un décès (événement) ait lieu dans un petit intervalle de temps, sachant que l'individu est vivant au temps t.

La fonction de survie représente la probabilité qu'un individu ait une durée de vie supérieure à t, et peut s'exprimer par :

$$S(t) =, P(T > t) \tag{1.2}$$

Si T est une variable aléatoire continue, la relation entre $\tau(t)$ *et* $S(t)$ *peut tre formule par* :

$$S(t) = e^{-\int_0^1 \lambda(u)du} \tag{1.3}$$

La fonction $\tau(t) = \int_0^1 \tau(u)$ *du est connue sous le nom de fonction de risque cumul.*

1.4 Estimateur non paramétrique de Kaplan-Meier

Les méthodes non paramétriques sont souvent préférables pour estimer la fonction de survie $S(t)$. Elles prennent mieux en compte la censure et la troncature, et elles donnent une meilleure adéquation. Kaplan et Meier (1958) ont proposé un estimateur très efficace de $S(t)$, nommé l'estimateur produit limite.
Si on considère $t1 < t2 << tD$, les temps de survie distincts de n individus, où au temps ti il y a di des événements et plus que Yi individus susceptibles de subir un événement, l'estimateur de la fonction de survie proposé par Kaplan-Meier est donné par :[4]

$$:\hat{S}(t) = \begin{cases} \prod_{t_i < t} [1 - \frac{di}{Yi}] & ,si \quad t < t_1 \\ 1 & ,si \quad t \geq t_1 \end{cases} \tag{1.4}$$

La quantité $\frac{i_d}{Y_i}$ estime la valeur de la fonction de risque $\tau(t)$ *pour* t=t_i.

L'estimateur produit limite est une fonction en escaliers qui effectue des sauts à chaque événement au temps k La grandeur du saut ne dépend pas uniquement du nombre d'événements au temps t_i mais aussi du nombre de censures à ce temps-là

1.4.1 Données de survie

La figure ci-dessous représente un exemple de courbe de survie de Kaplan-Meier des adultes sous traitement de COVID-19, à partir des données « COVID-19 du centre de santé ABASS NDAO sur les patients admis sous traitement de COVID-19 » décrites dans les pages suivantes.

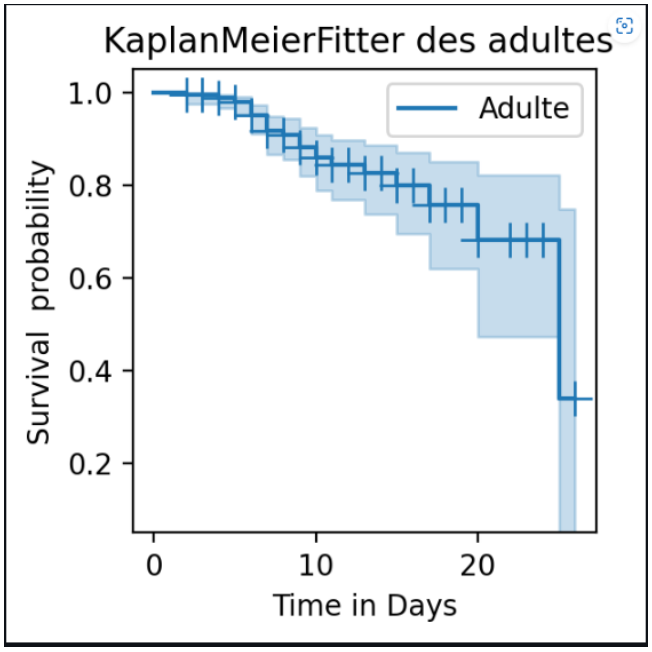


FIGURE 1.3 – Courbe de Kaplan-Meier de la fonction de survie $S(t)$.

1.5 Modèles de régression

1.5.1 Modèle semi-paramétrique de Cox

Les facteurs explicatifs ou les covariables sont fréquemment disponibles dans une telle étude, et nous sommes intéressés de savoir les effets de ces facteurs sur le changement du taux de risque à travers le temps.
Soit $Z = (Z_1, ...Z_p)$ un vecteur de covariables d'un individu ; ce vecteur serait composé de variables identifiant le traitement suivi, et de facteurs tels que l'âge d'un individu, sa pression systolique, etc... Le modèle de régression le plus connu est le modèle des risques proportionnels, qui spécifie le risque d'un individu avec covariable Z par :[7]

$$\lambda(t|z) = \lambda_0(t)e^{\beta^F z} \tag{1.5}$$

Les facteurs explicatifs ou les covariables sont fréquemment disponibles dans une telle étude, et nous sommes intéressés de savoir les effets de ces facteurs sur le changement du taux de risque à travers le temps.

Où $\beta = (\beta_1, ...\beta_p)^t$ est le vecteur des paramètres de régression.

Ainsi, la fonction de risque est le produit d'un terme dépendant du temps,
$\tau_0(t)$ nommé *risque* instantané *de base et d'un terme qui ne* dépend *que des covariables* $Z_1 e^{\beta^F Z}$.

Cox (1972) a suggéré une procédure qui permet d'estimer les paramètres, β sans faire de supposition sur la nature exacte de, $\tau_0(t)$. Ainsi, toute l'analyse se concentre sur les effets des covariables, d'où l'attribution du nom de modèle semi-paramétrique.[7]

Dans son article, Cox (1975) introduit la vraisemblance partielle basée sur des données qui ne demandent pas d'information sur, $\tau_0(t)$. Il écarte spécifiquement les temps de sur vie (ou d'événement) observés et le nombre d'événements à ces temps-là. En supposant que les censures sont indépendantes et non informatives, Cox écarte aussi les temps de censures et l'identité des individus associés à ces temps de censures.

La fonction de vraisemblance partielle est donc basée sur l'identité des individus susceptibles de subir un événement à chaque temps de survie (non censuré), le nombre des événements et l'identité des individus en risque à ce temps-là étant connus.

Cette fonction de vrai semblance prend la forme :

$$L(\beta) = \prod_{i \, g \, D} \frac{P(t \leq T < t + \Delta t \mid T \geq t)}{\sum_{j \in R_i} e^{\beta^F z_j}} \tag{1.6}$$

où D représente l'ensemble des indices des temps de survie (événements) observés, Z_i le vecteur des covariables pour les individus qui ont échoué au i^{me} temps de survie $t_(i)$ et R_i l'ensemble des individus qui sont en risque d'échouer au temps de survie $t_(i)$ (Hougaard, 1999).

1.5.2 Modèles paramétriques

Si la forme de $\tau_0(t)$ est précisée dans le modèle (1.5), on dira qu'il s'agit d'un modèle paramétrique. Parmi les modèles paramétriques, on mentionne le modèle dont le temps t suit une loi exponentielle caractérisée par un risque instantané constant (i. e. $\tau_0(t) = \tau$, avec une constante), et le modèle Weibull, caractérisé par :[8]

$$\lambda_0(t) = \alpha\lambda t^{\alpha-1}, \qquad \text{avec } \alpha > 0 \tag{1.7}$$

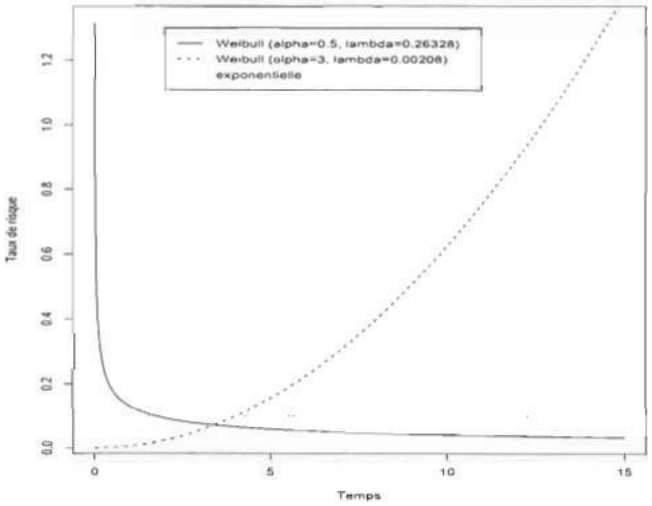


FIGURE 1.4 – CFonctions de risque de la loi exponentielle, et de la loi Weibull pour différentes valeurs de et .

1.5.3 Modèle de temps de la survie accéléré

En dépit de la grande popularité et de la polyvalence du modèle de Cox, il y a des bonnes raisons d'explorer des modèles alternatifs. Premièrement, l'hypothèse de la proportionnalité du modèle peut être non satisfaite pour certaines applications. Deuxièmement, il serait intéressant de trouver des modèles alternatifs qui caractérisent d'une façon différente l'association entre les Co variables et le temps de survie.

Une approche pour modéliser l'effet des Co variables est analogue à la régression linéaire classique. Dans cette approche, le logarithme du temps de survie $log(T)$ est modélisé par l'expression ;

$$\log(\mathrm{T}) = \beta' z + \epsilon \tag{1.8}$$

Où β est le vecteur de paramètres de régression inconnus, et $\in$ la variable d'erreur qui est indépendante du vecteur des covariables z.

Une transformation exponentielle de (1.8) mène à

$$T = e^{\beta^{\mathrm{F}} z} T_0 \tag{1.9}$$

Avec $T_0 = e^g$. Cette expression montre que le rôle de z est d'accélérer ou de ralentir le temps de survie. Ainsi, on dit qu'il s'agit d'un modèle de temps de survie accéléré. Le modèle (1.8) peut être spécifié par la fonction de risque :

$$\lambda(t|\, z) = \lambda_0 (te^{-\beta^F z}\ ) e^{-\beta^F z} \tag{1.10}$$

où $\tau_0(t)$. est la fonction de risque de T_0.

Un exemple de $\tau_0(t)$ est le risque instantané de base correspondant à la distribution de Weibull, i.e. $\tau_0(t) = \alpha\tau^{\alpha-1}$. (Klein et Moeschberger, 2003).

1.6 Processus de comptage et analyse de la survie

Au milieu des années 1970, Aalen présente sa théorie des martingales pour processus de comptage qui offre un cadre unifié pour les méthodes statistiques de l'analyse de la survie : Dans son travail, l'approche du processus de comptage utilise la représentation intégrale pour les statistiques des données censurées qui fournit une forme simple et unifiée des estimateurs, des statistiques de test, et des méthodes de régression. Ces méthodes utilisant les martingales permettent d'obtenir de simples expressions pour des statistiques compliquées, pour[10] des distributions asymptotiques de statistiques de test, et pour des estimateurs.

Dans l'approche du processus de comptage, la i^{me} observation est représentée par le couple $\{N_i(t), Y_i(t)\ (t > 0)\}$, avec :

$$\boldsymbol{N_i(t) = I\ (X_1 \leq t_1 \delta_i = 1)\ \ et\ \ \ Y_i\ (t) = I\ (X_i \geq t)} \tag{1.11}$$

Où $X_i est le minimum entre le temps de survie et le temps de censure, et \delta_i = 1$ l'indicateur de l'observation $\delta_i = 1$ si la donnée est non censurée et si $\delta_i = 0$).

Le processus continu à droite $N(t)$ est connu simplement sous le nom de processus de comptage, du fait qu'il compte le nombre des événements observés jusqu'au temps t ; et le processus continu à gauche $Y(t)$ est connu sous le nom de processus de risque, indiquant si un individu est susceptible de subir un événement au temps t.

Une illustration importante de l'approche du processus de comptage se résume dans l'étude des propriétés de l'estimateur de Nelson-Aalen $\wedge(t) et \wedge(t)$ (le risque cumulé). Le risque cumulé dans une région où il existe au moins une observation est :

$$\Lambda^*(t) = \int_0^t J(s)\bar{\lambda}(s)ds \tag{1.12}$$

Où $J(t) = I[\bar{Y}(t) > 0]$ est l'indicateur pour lequel au moins une observation est encore à risque. Avec :

$$\bar{Y}(t) = \int_0^t \frac{J(s)}{Y(s)} d\bar{N}(s) \text{ estime } \Lambda^*(t), \text{ ou } \bar{N}(t) = \sum_{i=1}^n N_I(t). \tag{1.13}$$

En effet,

$$\begin{aligned} \widehat{\Lambda}(t) - \Lambda^*(t) &= \int_0^t \frac{J(s)}{Y(s)} (d\bar{N}(s) - \bar{Y}(s)\bar{\lambda}(s)ds) \\ &= \sum_{i=1}^n \int_0^t \frac{J(s)}{Y(s)} dM_1(s), \text{ où } \bar{N}(t) = \sum_{i=1}^n Y_I(t). \end{aligned} \tag{1.14}$$

Où la martingale $M_i(s) = N_i(s) - \int_0^s Y_i(u)\delta(u)du$ (spécifique du $i$éme individu) a une espérance égale à zéro.

La martingale $M_i(s)$ représente la différence entre le nombre d'événements observé, $N_i(s)$), et le nombre d'événements prédit par le modèle pour le $i$éme individu.

Chapitre 2

Application de l'analyse de survie au cas de l'hopital ABASS NDAO

2.1 Présentation du jeu de données

Pour faire notre étude, nous disposons d'un jeu de données qui contient les informations des patients hospitalisés sous COVID-19 à l'hôpital ABASS NDAO de Dakar. Il est composé 332 observations pour 41 variables dont 9 variables qualitative, 31 quantitatif et une variable cible qualitative qui indique si le patient est décédé ou pas.

Pour effectuer une analyse de survie de nos données nous avons seulement besoins des variables suivantes

- **Sexe**
- **Tranche d'âge**
- **EVENEMENT**

2.2 Choix de l'outil d'analyse de survie

En python, il existe actuellement deux librairies principales pour faire l'analyse de survie. La première s'appelle 'lifelines' et la deuxième 'scikit-survival'.

Pour effectuer une analyse de survie de nos données nous avons seulement besoins des variables suivantes

- **Lifelines :** est dédié à l'analyse de survie classique
- **Scikit-survival** vise surtout l'apprentissage automatique (le machine learning) dans le cas de données survie.

Si on débute avec l'analyse de survie, vaut mieux de commencer avec 'Lifelines'. Il est un peu plus facile à installer et à prendre en main et sous-entends que l'on maîtrise déjà assez bien Python et aussi le package 'scikit-learn' de l'apprentissage automatique. Il est un peu plus avancé. Les deux librairies permettent de créer la courbe de survie de Kaplan-Meier. Dans la suite de notre travail nous allons faire une comparaison de ces deux librairies et utiliser celle qui nous offre les informations pour réaliser analyse de survie.

Dans la suite de notre travail nous avons décidé d'utiliser Lifelines puisqu'il nous offre plusieurs options d'affichage que scikit-survival ne peut pas nous offrir.

On peut également préciser certaines options, par exemple :

— **Ci show :** indique si on veut afficher l'intervalle de confiance. J'active cette option.
— **Show censor :** contrôle l'affichage des points censurés avec des petites croix sur la courbe. Je l'active aussi. On peut également donner un label et d'autres options classiques pour un 'plot' dans 'matplotlib'. J'ajoute également quelques options pour l'annotation des axes et la commande show pour afficher la figure obtenue.

En analyse de survie il est très important pour nous de savoir quel facteur affecte le plus la survie. Donc, dans la suite, nous discutons de l'estimateur Kaplan-Meier basé sur divers groupes.

2.3 Estimateur de Kaplan-Meier basé sur divers groupes

2.3.1 Transformation des variables nécessaire pour l'analyse de survie

Dans la partie encodage des variables nécessaire pour faire notre analyse de survie nous avons décidé d'ordinariser nos variables puisque la classe KaplanMeierFitter prend en entrée des entiers.

2.3.2 DIVISION, Affichage de la liste complète de survival-probability et de la courbe de survie

Après la partie encodage nous allons par la suite organiser et divisé nos données en cinq groupes : hommes, femmes, adultes, personnes âgées, adolescent.

Notre objectif ici est de vérifier s'il existe une différence significative dans le taux de survie si nous divisons notre ensemble de données en fonction du sexe et de la tranche d'âge.

Pour ce faire :

J'utilise la librairie 'pandas' pour importer les données dans un dataframe.

Depuis cette librairie, j'importe la classe KaplanMeierFitter. Tout d'abord, nous avons créé cinq objets de la classe que nous avons nommé kmf-m pour le groupe des hommes, kmf-f pour le groupe des femmes, kmf-adu pour le groupe des adultes, kmf-persage pour le groupe des personnes âgée et kmf-ado pour le groupe des adolescents.

Les objets sont initialisés, mais, pour l'instant ils sont vides.

Nous allons ajuster les données divisées en groupe dans nos objets, chaque groupe ajuster dans un objet qui lui est spécifique.

La méthode fit () permet d'entraîner le modèle en indiquant les données des durées de survie (dans la colonne $dure_hosp$) et la censure correspondante dans la colonne EVENEMENT.

La méthode fit () calcule l'estimation de la fonction de survie et d'autres statistiques qui peuvent être intéressantes. Par exemple, l'intervalle de confiance à $95pourcent$ qu'on a besoin d'afficher parfois sur la figure.

En utilisant la méthode $survival_function$, nous avons généré la liste complète des probabilités de survie au cours du temps pour chaque groupe selon le sexe et la tranche d'âge.

Enfin nous avons tracé les courbes de survie selon les différents groupes.

2.3.3 Liste des probabilités de survie au cours du temps selon le sexe

2.3.3.1 Liste des probabilités de survie au cours du temps pour les Hommes

Time	Hommes
0	1
3	1
4	1
5	0.9922
6	0.9396
7	0.8978
8	0.8978
9	0.8584
10	0.8329
11	0.8077
12	0.8077
13	0.7755
14	0.7395
15	0.7395
16	0.7395
17	0.6827
18	0.6827
19	0.6827
20	0.6827
22	0.6827
23	0.4552
25	0

FIGURE 2.1 – Liste des probabilités de survie au cours du temps pour les Hommes

Par exemple, pour les hommes, la probabilité de survie est égale à 0.9633 pour t=6 jours

2.3.3.2 Liste des probabilités de survie au cours du temps pour les Femmes

Time	Femmes
0	1
2	0.9944
3	0.9944
4	0.9775
5	0.9711
6	0.9633
7	0.9434
8	0.931
9	0.8893
10	0.8722
11	0.8722
12	0.8722
13	0.8373
14	0.8373
15	0.7933
16	0.7933
17	0.7272
18	0.7272
19	0.7272
20	0.5817
22	0.5817
23	0.5817
25	0.5817

FIGURE 2.2 – Liste des probabilités de survie au cours du temps pour les Femmes

Par exemple, pour les Femmes, la probabilité de survie est égale à 0.9633 pour t=6 jours

2.3.4 Liste des probabilités de survie au cours du temps selon la tranche d'âge

2.3.4.1 Liste des probabilités de survie au cours du temps pour les Adultes

Time	Adultes
0	1
2	0.9364
3	0.9364
4	0.9891
5	0.9804
6	0.9499
7	0.9172
8	0.9004
9	0.8821
10	0.859
11	0.8452
12	0.8452
13	0.8256
14	0.8256
15	0.7933
16	0.7932
17	0.7577
18	0.7577
19	0.7577
20	0.0815
22	0.0815
23	0.0815
24	0.3405
25	0.3406

FIGURE 2.3 – Liste des probabilités de survie au cours du temps pour les Adultes

Par exemple, pour les Adultes, la probabilité de survie est égale à 0.9499 pour t=6 jours

2.3.4.2 Liste des probabilités de survie au cours du temps pour les Personne-âgées

Time	Personne-âgées
0	1
3	1
4	0.9815
5	0.9815
6	0.9592
7	0.9358
8	0.8589
10	0.8294
11	0.8294
12	0.8294
13	0.7258
14	0.6221
17	0.5184
18	0.5184
19	0.5184
18	0.7272
19	0.7272
23	0

FIGURE 2.4 – Liste des probabilités de survie au cours du temps pour les Personne-âgées

Par exemple, pour les Personne-âgées, la probabilité de survie est égale à 0.9592 pour t=6 jours

2.3.4.3 Liste des probabilités de survie au cours du temps pour les Adolescents

Time	Adolescents
0	1
8	1

FIGURE 2.5 – Liste des probabilités de survie au cours du temps pour les Adolescents

Par exemple, pour les Adolescents, la probabilité de survie est égale à 1 pour t=8 jours

2.4 Courbe de survie de kaplanMeier

Maintenant il nous reste à afficher la courbe obtenue. Nous avons créé cinq figures à l'aide du package 'matplotlib' et puis nous les attachons des axes, en précisant les coordonnées ou ces axes doivent être placés dans la figure. La commande 'plot' de l'objet 'kmf' ajoute la courbe de survie sur les axes.

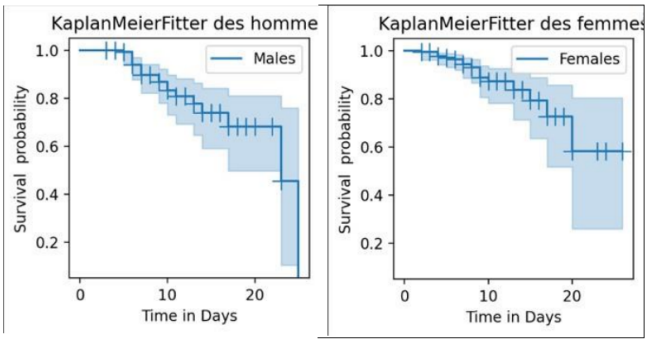


FIGURE 2.6 – Courbe de kaplan Meier selon le sexe

Analyse :

— **Pour les hommes**
On voit que les marches d'escalier sont petites, surtout au début, parce que le nombre de patients observés est élevé, on voit aussi que la courbe descend jusqu'à la probabilité 0 cela veut dire qu'un pourcentage de patients décédé complètement du Covid19.

La médiane de survie pour les hommes qui correspond au temps pour lequel la probabilité de survie est égale à 50pourcents. Vaut 24jours.

D'après ces données plus de 50pourcents de patients ont été guéris de la maladie et ont vécu plus de 15 jour après leur diagnostic de Covid 19 cela explique l'existence de beaucoup de point de censure marqué par des croix 0 à 15eme jour.

— **Pour les femmes**
On voit que les marches d'escalier sont petites, surtout au début, parce que le nombre de patients observés est élevé, on voit aussi que la courbe ne descend

pas jusqu'à la probabilité 0 cela veut dire qu'un pourcentage de patients guérît complètement du Covid19.

La médiane de survie pour les femmes qui correspond au temps pour lequel la probabilité de survie est égale à 50pourcents. Vaut 20jours.lité de survie est égale à 50pourcents. Vaut 24jours.

D'après ces données plus de 50pourcents de patients ont été guéris de la maladie et ont vécu plus de 15 jours Après leur diagnostic de Covid 19 cela explique l'existence de beaucoup de point de censure marqué par des croix 0 à 15eme jour.

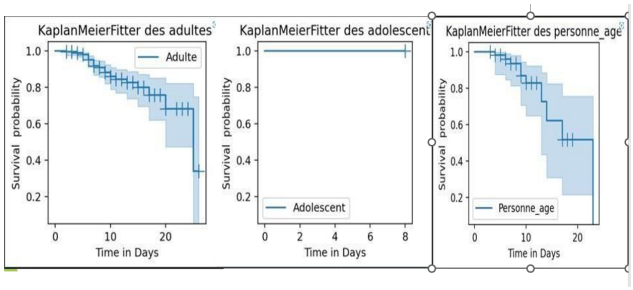


FIGURE 2.7 – Courbe de kaplan Meier selon la tranche d'âge

Analyse :

— **Pour les adultes**
On voit que les marches d'escalier sont petites, surtout au début, parce que le nombre de patients observés est élevé, on voit aussi que la courbe ne descend pas jusqu'à la probabilité 0 cela veut dire qu'un pourcentage de patients guérît complètement du Covid19.

La médiane de survie pour les adultes qui correspond au temps pour lequel la probabilité de survie est égale à 50pourcents. Vaut 28jours.

D'après ces données plus de 50pourcents de patients ont été guéris de la maladie et ont vécu plus de 15 jour après leur diagnostic de Covid 19 cela explique l'existence de beaucoup de point de censure marqué par des croix 0 à 15eme jour.

— **Pour les adolescents**
On voit que la courbe ne descend même pas cela veut dire qu'un pourcentage de patients adolescent guérit complètement de la covid19.

D'après ces données plus de 50pourcents de patients ont été guéris de la maladie et ont vécu plus de 15 jour après leur diagnostic de Covid 19 cela explique l'existence de beaucoup de point de censure marqué par des croix 0 à 15eme jour.

— **Pour les personnes âgées**
On voit que les marches d'escalier sont petites, surtout au début, parce que le nombre de patients observés est élevé, on voit aussi que la courbe descend jusqu'à la probabilité 0 cela veut dire qu'un pourcentage de patients décédé complètement du Covid19.

La médiane de survie pour les personnes âgées qui correspond au temps pour

lequel la probabilité de survie est égale à 50pourcents. Vaut 16jours.

D'après ces données plus de 50pourcents de patients ont été guéris de la maladie et ont vécu plus de 8 jour après leur diagnostic de Covid 19 cela explique l'existence de beaucoup de point de censure marqué par des croix 0 au 8eme jour.

2.5 Conclusion sur les analyses

Nous remarquons que la COVID-19 affecte plus les hommes surtout les plus âgées que les femmes et d'habitude les personnes âgées hospitalisé plus de 22 jours finissent par mourir.

Nous remarquons aussi que la maladie a un faible pourcentage de tuer un enfant ou un adolescent, nous pouvons même dire que ces derniers résistent à la maladie et ces derniers même s'ils sont admis guérissent au plus tard le 8émé jours après leurs débuts d'hospitalisations.

Chapitre 3

GENERALITES SUR L'APPRENTISSAGE AUTOMATIQUE

3.1 Définition du Machine Learning

De manière générale, un programme informatique permet de résoudre un problème dont la solution est connue (calculer la moyenne générale des étudiants, gestion d'une boutique). Cependant, pour certains problèmes, la solution exacte nous est inconnue (la reconnaissance faciale, jouer contre un être humain...). Écrire des programmes simples, permettant de traiter ces problèmes, est impossible. Toutefois, il est très facile d'avoir une base de données regroupant de nombreuses instances du problème considéré. L'apprentissage automatique, encore appelé Machine Learning, est l'étude des algorithmes permettant de résoudre automatiquement un problème considéré à l'aide de données. En effet les algorithmes d'apprentissage automatique construisent des modèles basés sur des instances de données appelées données d'apprentissage. Le terme d'apprentissage automatique (AA) a été inventé en 1959 par Arthur Samuel (un IBMer américain et pionnier des jeux informatiques). Le Machine Learning est une sous partie de l'intelligence artificielle. Ce dernier est la science quiétudie l'intelligence humaine par sa modélisation et sa stimulation au moyen de programmes informatiques.[11]

De nos jours l'apprentissage automatique a deux principaux objectifs, l'un est de classer les données en fonction des modèles qui ont été développé, l'autre est de faire de la prédiction L'apprentissage automatique est à la croisée de plusieurs disciplines à savoir :

3.2 Les méthodes d'apprentissage

Le Machine Learning est la capacité d'apprendre à une machine à prendre des décisions de manière automatique. Ainsi, pour apprendre, la machine a besoin de Datasets ou jeux de données. Les Datasets sont formés de features ou variables d'entrée et de targets ou variables de sorties. Chaque individu appartenant aux Datasets représente un Sample. Il existe plusieurs méthodes d'apprentissage automatiquement à partir des données dépendamment des problèmes à résoudre et des données disponibles. Généralement on peut classer ces approches en 4 grandes catégories :

De nos jours l'apprentissage automatique a deux principaux objectifs, l'un est de classer les données en fonction des modèles qui ont été développé, l'autre est de faire de la prédiction L'apprentissage automatique est à la croisée de plusieurs disciplines à savoir :

3.2.1 L'apprentissage automatique supervisé

3.2.1.1 Définition

Le Machine Learning avec supervision est une technologie élémentaire mais stricte. Pour ce type de méthode d'apprentissage, le système dispose de données bien étiquetées. Une donnée étiquetée est une donnée constituée d'un ensemble d'exemples d'apprentissage, où chaque exemple est une paire composée d'une valeur d'entrée (x) et d'une valeur de sortie (y) souhaitée. La machine grâce à ses labels d'entrée va pouvoir faire des recherches de solution pour obtenir en fonctions de ces derniers des labels de sorties. Le but recherché est que l'ordinateur apprenne la règle générale qui mappe les entrées et les sorties. En effet, Le Machine Learning avec supervision peut être utilisée pour faire des prédictions sur des données indisponibles ou futures (on parle de « modélisation prédictive »). L'algorithme essaie de développer une fonction qui prédit avec précision la sortie à partir des variables d'entrée.[6]

3.2.1.1.1 Les familles d'apprentissage supervisé

L'apprentissage supervisé peut être divisé en deux grandes familles qui sont :

L'apprentissage supervisé peut être divisé en deux grandes familles qui sont :

— **Apprentissage supervisé symbolique :**
L'apprentissage supervisé symbolique est une méthode inspirée de l'intelligence artificielle et dont les fondements reposent beaucoup sur des modèles de logique, sur une représentation binaire des données (vrai / faux) et sur les méthodes de représentation des connaissances.

— **Apprentissage supervisé numérique**
L'apprentissage supervisé numérique est une méthode inspirée de la statistique. Les données sont en général des vecteurs de réels et les méthodes font intervenir des outils provenant des probabilités, de l'algèbre linéaire et de l'optimisation.

Soient :

— X $ensembledesentres(connusetfixe)$
— Y *ensemble des sorties*
— (x_n, y_n) *couple entres et sortie* , tq $n \in$, $x_n \in$ X , $y_n \in$ Y ,
$y_n = f(x_n) + w_n$
avec w_n un bruit de mesure

Une base de données d'apprentissage est l'ensemble du couple (x_n, y_n) tiré d'une loi $X * Y$ fixe (x_n) et inconnue (y_n).

La méthode d'apprentissage supervisée utilise ce type de base d'apprentissage afin de déterminer une estimation f notée g (tq une nouvelle entrée x on l'associe à $g(x)$) est appelée fonction de prédiction.

Il existe deux types de sous-problèmes en apprentissage supervisé numérique qui sont :

— **La classification :**
on parle de classification lorsque la variable de sortie est discrète (la variable de prédiction est non numérique), $Y = *1.2$, ... , $I +$. En effet pour ce type d'apprentissage supervisé, la variable d'entrée est attribuée à une classe ou

étiquette. Donc il s'agit dans un premier temps de classer les données d'entrée (phase d'étiquetage) et ensuite de prédire la classe de notre nouvelle donnée reçue à partir des classes déjà apprises. La fonction de prédiction est appelée classifieur.

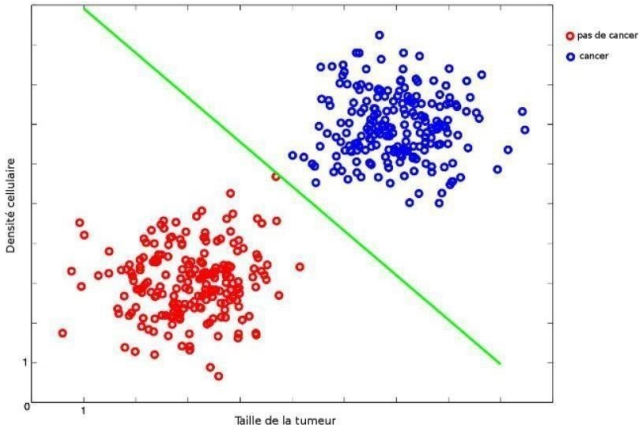


FIGURE 3.1 – Exemple classification

— **La régression :**
on parle de régression lorsque la variable de sortie est continue (valeur de prédiction numérique), ∈. La variable de sortie est une valeur dans un ensemble continu de réels. La fonction de prédiction est alors appelée régresser.[9]

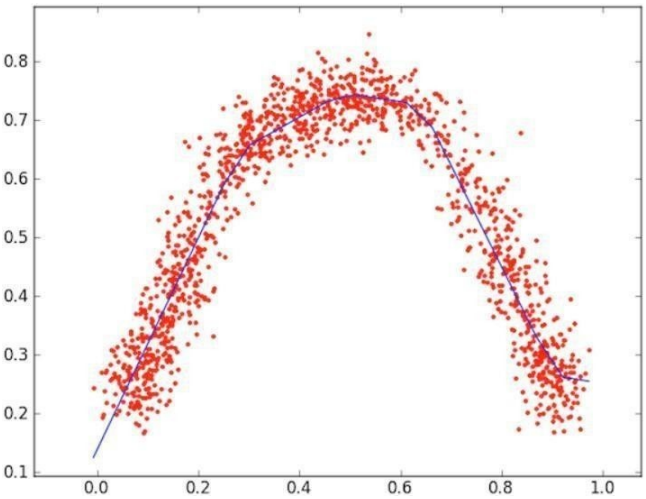


FIGURE 3.2 – Exemple de régression

3.2.2 Apprentissage non supervisé

Pour ce type d'apprentissage, la base de données d'apprentissage utilisé ne possède que d'un ensemble de données en entrée auquel l'algorithme se base sa base pour découvrir leur structure. Il y'a aucune information sur les données de sorties. Cette technique d'apprentissage est utilisée pour partitionner en groupe d'éléments homogènes les données d'entrée.[11]

On peut donc tout simplement retenir que, l'apprentissage non-supervisée consiste à caractériser des samples issus de notre Datasets sans pour autant connaitre les targets. En outre l'algorithme d'apprentissage non-supervisée permettra d'établir le nombre de catégories distinct de notre Datasets mais ne pourrons pas prédire leur classe, en effet il va essayer de faire une prédiction en lui associant d'autre structure similaire.[3]

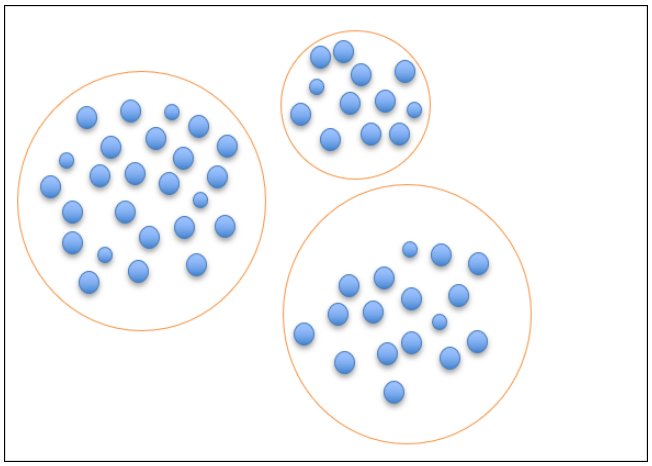


FIGURE 3.3 – Exemple d'apprentissage non supervisé

Ainsi le principal objectif d'un algorithme d'apprentissage supervisé est de déterminer des résultats généralisés des données d'entrées inconnues grâce aux comportements apprises des données étiquetées.

3.2.3 Apprentissage par renforcement

L'apprentissage par renforcement est généralement utilisé en théories des jeux. Pour ce type d'apprentissage, le programme s'exécute dans un Dataset dynamique ou il doit atteindre un certain but. Ainsi, le programme apprenti permet à un individu placé dans notre environnement, pouvant y effectuer des opérations de recevoir des retours sous formes de récompense ou de punition, pour optimiser son gain. Le programme apprend alors par lui-même un modèle capable de prendre la meilleure décision étant donné un état de l'environnement.

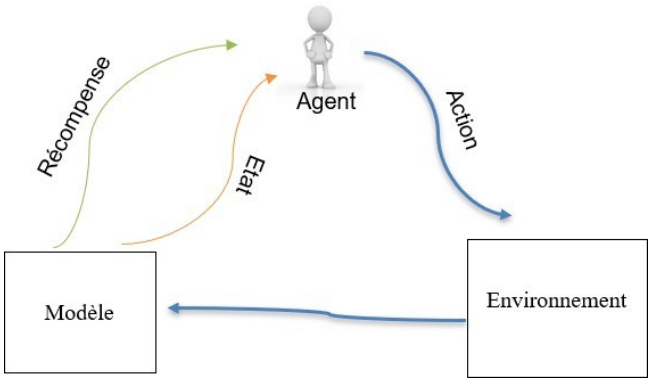
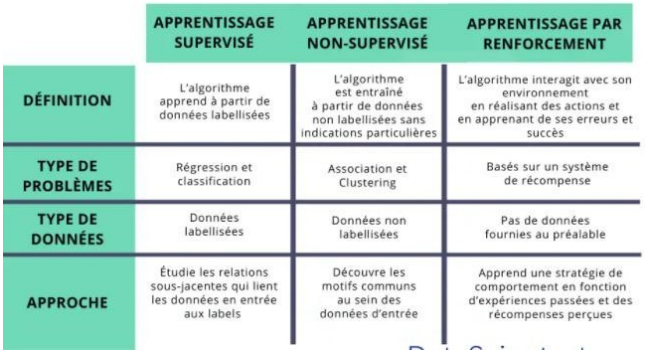


FIGURE 3.4 – Schéma descriptive de l'apprentissage par renforcement

3.2.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé est un mixte ente l'apprentissage supervisé et l'apprentissage non supervisé. Ce type d'apprentissage utilise à la fois des données étiquetées et des données non étiquetés. Cette approche est très avantageuse. En effet l'apprentissage semi-supervisé ne nécessite que quelques données étiquetées ce qui est très économique en termes de cout et de de gain de temps. De plus le fait de jouer entre deux types de données différentes améliore la performance de notre model final de prédiction.[3]

Pour notre projet, disposant des données qualitatives, nous allons dans ce cas orienter nos recherches sur les différents algorithmes de classifications de la méthode d'apprentissage supervisé.



	APPRENTISSAGE SUPERVISÉ	APPRENTISSAGE NON-SUPERVISÉ	APPRENTISSAGE PAR RENFORCEMENT
DÉFINITION	L'algorithme apprend à partir de données labellisées	L'algorithme est entraîné à partir de données non labellisées sans indications particulières	L'algorithme interagit avec son environnement en réalisant des actions et en apprenant de ses erreurs et succès
TYPE DE PROBLÈMES	Régression et classification	Association et Clustering	Basés sur un système de récompense
TYPE DE DONNÉES	Données labellisées	Données non labellisées	Pas de données fournies au préalable
APPROCHE	Étudie les relations sous-jacentes qui lient les données en entrée aux labels	Découvre les motifs communs au sein des données d'entrée	Apprend une stratégie de comportement en fonction d'expériences passées et des récompenses perçues

FIGURE 3.5 – Tableaux comparatifs de méthodes d'apprentissage supervisé, non supervisé et par renforcement

3.3 Quelques méthodes d'apprentissage supervisé pour la classification

Nous voulons résoudre un problème de classification parce que la variable à prédire est une variable qualitative.[11]

3.3.1 Arbres décisionnels

L'apprentissage par arbre des décisions est une méthode classique en apprentissage automatique. Il permet de construire un modèle sous forme d'arbre depuis un ensemble d'apprentissage, qui prédit la valeur d'une nouvelle variable d'entrée bien précise des variables d'entrée déjà existante.
L'arbre est constitué :

- **Une racine :**
  c'est le début de l'arbre. La racine contient ainsi l'ensemble des observations.
- **De branches :**
  les séries de branche permettent de faire un choix.
- **De Nœuds :**
  Chaque intersection de branche forme un nœud. Un nœud est une règle bien précise. Parcourir l'arbre revient donc à vérifier une série de règles. Chaque nœud doit diviser le mieux l'ensemble des observations.[9]
- **Les feuilles :**
  elles représentent les classes à prédire.

Exemple :
Le tableau ci-dessous représente des observations faites sur la taille des smartphone réparties en trois grandes espèces.

Observation	Longueur	Largeur	Espèce
1	2.7 cm	40 cm	A
2	3.6 cm	16 cm	C
3	2.4 cm	21 cm	B

FIGURE 3.6 – Observations sur des smartphones

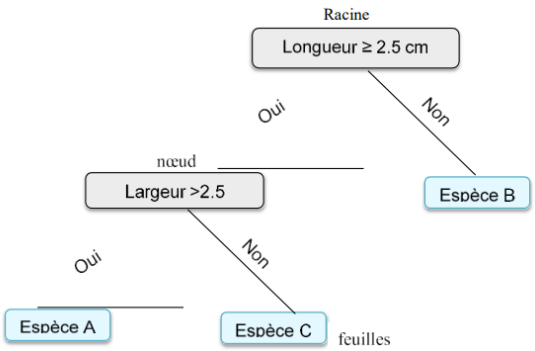


FIGURE 3.7 – Exemple d'arbre décisionnel

Nb : Pour cet exemple, l'arbre à une profondeur de 2. La profondeur d'un arbre correspond au nombre total de nœud dont il dispose plus la racine.

3.3.2 La méthode des K plus proche voisins

En abrégés K-NN ou KNN, la méthode des K plus proche voisins est un algorithme de reconnaissance des formes. Elle est une méthode non paramétrique et très utilisée pour la classification.
Ainsi, on dispose d'un Dataset de :

N couple tq N = (X, Y) avec X L'ensemble des x_n et Y l'ensemble des y_n
soit k le nombre d'échantillon d'apprentissage avec k $\in$ *et¸trs petit*

Pour prédire la valeur de sortie y_i d'une nouvelle entrée x_i, la méthode consiste à sélectionner en compte les k échantillon d'entrée dont les k sont plus proche de la nouvelle entrée, et ensuite retenir la classe majoritairement représenter. Les entrés les plus proche sont mesurée grâce à la distance de recouvrement (ou distance Hamming). k est une valeur donnée par l'utilisateur.
Exemple :

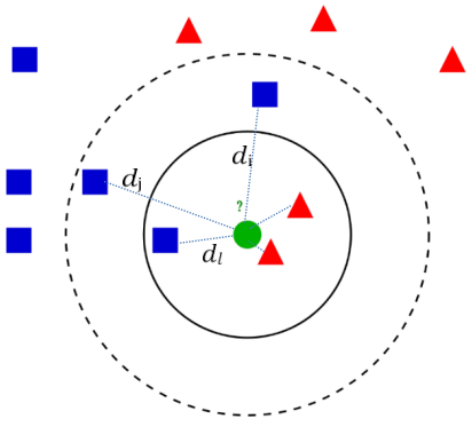


FIGURE 3.8 – Exemple de classification K- NN

Dans cet exemple on a des samples reparties en 2 classes (carré bleu et triangle rouge). En vert nous avons une nouvelle entrée x. L'échantillon de test peut soit être classé dans le cercle en ligne pointillé si $k = 5$. Dans ce cas la classe majoritairement représentée sera le carré bleu ou dans le cercle en ligne pleine si $k = 3$ dans ce cas la classe majoritairement représentée sera le triangle rouge.[11]

Où d_k *reprsente la distance distance entre notre nouvelle entre et les entres dja existante.*

3.3.3 Les machines à vecteur de support

La machine à support vectorielle encore appelé séparateur à vaste marge est une technique d'apprentissage supervisé permettant de résoudre des problèmes de la discrimination. Ce problème repose à trouver une droite, appelée frontière séparatrice optimale, à partir d'un échantillon de données. La frontière de séparation est la droite qui maximise la marge. On appelle marge la distance entre la frontière de séparation et les échantillons de données.

— **Description de l'algorithme de SVM :**
Soient :

Un espace de dimension N (N étant le nombre d'attribut de données)

$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_k, y_k), ... (x_p, y_p)\} <$ ^{N} $\times$ *$-1,1+$

où y_k sont les labels, p est la taille de l'ensembled'apprentissage.

soit $h(x)$ une fonction tq $y = h(x) =$ $(w^T x + w_0)$

où w est un vecteur de poids tq $w = (w_0, ..., w_N)^T$

si $h(x) \geq 0$ alors x $\in$ classe 1 sinon x $\in$ classe -1

si $h(x) = 0$ alors $h(x)$ est un hyperplan

FIGURE 3.9 – Description de l'algorithme de SVM :

La distance d'un échantillon x_k à l'hyperplan est donnée par sa projection orthogonale sur le vecteur poids :

$$dist(x_k; (w, w_0)) = \frac{| w^T x_k + w_0|}{||w||} +$$

La marge correspond à la distance entre les échantillons les plus proches et l'hyperplan

$$dist(w, w_0) = \min dist(x_i, (w, w_0))$$

L'hyperplan séparateur (w, w_0)de marge maximale est donnée par :

$$\arg \{ \begin{matrix} w^T x^+_{marge} + w_0 = 1 \\ w^T x^-_{marge} + w_0 = -1 \end{matrix}$$

$$d'où \star k \in *1,2, ..., p+, \qquad y_k(w^T x_k + w_0) \geq 1$$

Exemple
Un individu est à la recherche d'une bonne destination de voyage pour ses prochaines vacances. Une ville est représentée par la densité de la population et la température. Donc on aura un espace à deux dimensions. Les données d'apprentissage sont des villes déjà visité. Elles sont classées en deux grand groupe qui sont : en + pour les villes aimé par l'individu et les villes qui ont été moins apprécié.[2]

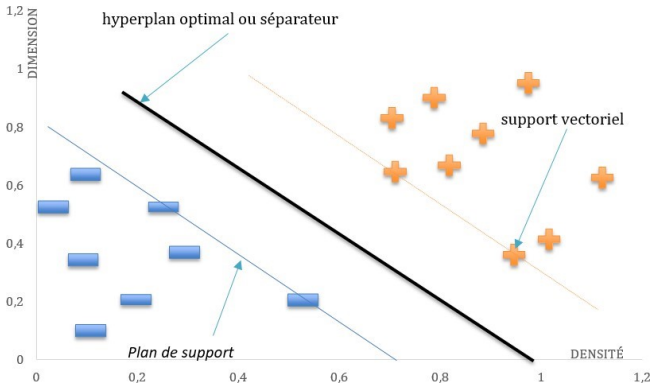


FIGURE 3.10 – Exemple de SVM de dimension

3.3.4 Les réseaux de neurones

Les réseaux de neurones encore appelé réseaux de neurones artificiels (RNA) sont des techniques d'apprentissage automatique qui se basent sur le fonctionnement d'un neurone du cerveau humain. En M.L. le neurone est une unité qui est généralement par une fonction sigmoïde :[9]

$$f(x) = \frac{1}{1 + e^{-x}}$$

Les réseaux de neurones sont des techniques très performantes pour des taches de machine Learning et leur domaine d'application sont vaste et diversifier parmi eux on peut citer :

- **La reconnaissance d'image**
- **Les classifications de textes ou d'images**
- **Indentifications d'objets**
- **Filtrage d'un set de données**

3.3.5 Les méthodes d'ensembles

Les méthodes d'ensemble sont des méthodes d'apprentissage qui repose sur la combinaison de plusieurs méthodes simple ou unique de base d'apprentissage automatique (voir les algorithmes vus ci-dessous). La combinaison de plusieurs algorithmes d'apprentissage permet d'obtenir de meilleures prédictions (robustesse de 'algorithme, efficacité et fiabilité). Les méthodes d'ensembles peuvent être réparties en deux groupes qui sont : les méthodes d'ensembles parallèles et les méthodes d'ensembles séquentielles .[5]

3.3.5.1 Méthodes d'ensembles parallèles

3.3.5.1.1 Le boostrap aggregating

Le boostrap aggregating encore appelé bagging est un méta-algorithme de Machine Learning permettant d'améliorer la stabilité et la précision des algorithmes d'apprentissage de base. Il réduit la variance (mesure de la dispersion des valeurs d'un échantillon ou d'une distribution de probabilité). Le bagging peut être appliqué avec n'importe quel type de modèle. Il est généralement appliqué avec l'arbre de décision.[12]

— **Description de l'algorithme :**
Nous avons à la base un ensemble d'enregistrement D de taille n. Tout d'abord l'algorithme découpe cet ensemble en sous ensemble D_i de taille n' puis en sélectionne certain afin de pouvoir faire des sondages avec. Ce sondage permettra d'estimer les caractéristiques de D. Si $n' = n$ alors pour n plus grand, l'ensemble $D_i -> 1 - \frac{1}{e}$

d'exemple unique de D, le reste sont considéré comme des doublons. Ensuite m modèles sont entrainer à l'aide de m échantillons de boostrap. Et enfin, la prédiction est obtenue par vote de majorité des m modèles.

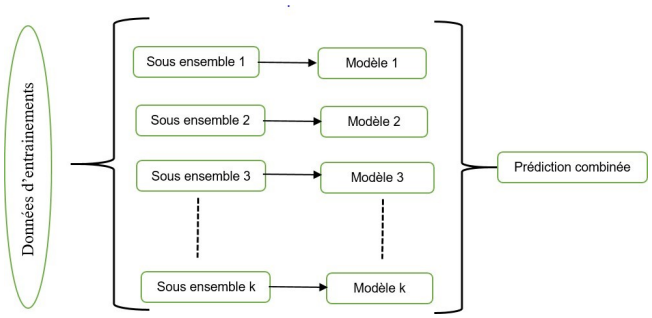


FIGURE 3.11 – Général du fonctionnement des méthodes d'ensembles parallèles

3.3.5.1.2 La forêt aléatoire

La forêt aléatoire ou Radom Forest est un algorithme qui combine le concept des sous espaces aléatoire (arbres décisionnels) et le bagging. En effet, cet algorithme effectue un apprentissage sur plusieurs arbres décisionnels autrement dit, l'ensemble des propositions d'arbre décisionnel forme une forêt aléatoire. Les différentes étapes de construction d'une forêt sont :

— **Prendre X nombre d'enregistrement du jeu de données**
— **Soit M l'ensemble de variable d'entrée. On prend un nombre K dans M**
— **On entraine un arbre de décision sur ces données**
— **On répète N fois le processus de sorte à obtenir N arbre de décision**
— **L'ensemble des arbres de décisions forme la forêt**
Ainsi chaque arbre propose une classe différente. La classe retenue sera celle qui sera le plus représenté parmi l'ensemble des arbres de la forêt.[6]

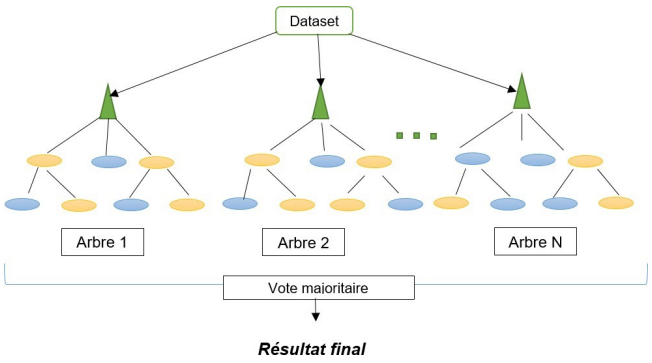


FIGURE 3.12 – : Exemple de Random Forest

3.3.5.2 Méthodes d'ensembles séquentielles

La méthode d'ensemble séquentielle est une méthode qui permet de regrouper sur la combinaison d'algorithme de classifier binaire. Cette méthode peut produire plusieurs méthodes ayant chacun une capacité de classifier. Chaque modèle est appelé classifieur faible. Ainsi on appelle classifieur faible tout modèle permettant de mieux classifier que s'il était aléatoire. Par itération successive, chaque classifieur faible est ajouté au classifieur final. Un apprenant faible est un algorithme qui fournit des classifieurs faibles, tant qu'ils proposent une performance au moins un peu supérieure à celle d'un classeur aléatoire. Les données mal classées sont boostées pour qu'elle aille plus d'importance vis-à-vis de l'apprenant faible au prochain tour.

3.3.5.2.1 L'algorithme AdaBoost

AdaBoost est un méta-algorithme de boosting utilisé afin d'améliorer les performances des algorithmes d'apprentissage. Son principe repose sur la sélection itérative de classifieur faible en fonction d'une distribution des exemples d'apprentissage (pondéré en fonction de la difficulté avec le classifieur courant). Chaque exemple est pondéré en fonction de sa difficulté avec le classifieur courant.

3.3.5.2.2 L'algorithme de gradient boosting machine

Le gradient boosting machine (GBM) est un méta-algorithme similaire à celui d'AdaBoost. Cependant, le GBM produit un modèle de prédiction dans le seul but de renforcer les modèles de base d'algorithme ayant une prédiction faible afin de minimiser les erreurs. On parle de prédiction faible, lorsque qu'on un écart entre la réalité et les valeurs de prédiction. Il est généralement utilisé avec les arbres de décision. Il existe une variété d'implémentation de GBM. Parmi eux on peut citer Xboost (extrem Gradient Boosting) qui utilise des approximations plus précises pour trouver les meilleurs modèles de base. D'ailleurs c'est le modèle le plus utilisé par la communauté des dataScientists pour son efficacité et ses merveilleuses performances.

Chapitre 4

Présentation et préparation des données

4.1 Présentation des données

Nous utilisons le même jeu de donnée que pour l'analyse de survie mais cette fois ci nous allons prendre en compte les autres variables .
Pour faire notre étude, nous disposons d'un jeu de données qui contient les informations des patients hospitalisés sous COVID-19 à l'hôpital ABASS NDAO de Dakar. Il est composé 332 observations pour 41 variables dont 9 variables qualitative, 31 quantitatif et une variable cible qualitative qui indique si le patient est décédé ou pas.

Ainsi pour chaque patient nous disposant des données suivantes :

- **ADMISSION :**
  Date début hospitalisation
- **SEXE :**
  Correspond au sexe des patients avec deux types de genre M ou F
- **AGE :**
  Correspond aux âges des patients
- $Tranched'Age$:
  composé de deux types : Adultes et Personne Âgée
- **Test PCR :**
  composé de deux types : Adultes et Personne Âgée
- **Test PCR :**
  Le test nasopharyngée PCR permet de déterminer au moment du prélèvement si vous êtes porteur du virus de la Covid-19 et constitue le test de référence le plus sensible pour le diagnostic de l'infection. Avec le test pcr deux types de résultats existent soit positif ou faux négatif.
- **Etat Diabète :**
  Composé de trois résultats : Non, Diabète inaugural, DT2
- **Diabète :**
  Composé de deux résultats soit oui ou non
- **Coomorbi :**
  Le terme médical "comorbidité" est régulièrement prononcé avec l'épidémie de Covid-19. Pour cause, la présence de comorbidité est un des deux facteurs de risque de formes graves les plus importants, après l'âge. La présence de comorbidités a priorisé l'accès à la vaccination pour certaines personnes et

l'administration d'une troisième dose. Composée de deux résultats 1 qui signifie qu'il Ya Coomorbi et 0 qui signifie le contraire.

— **HTA :**
correspond à une augmentation anormale de la pression du sang sur la paroi des artères. Composée de deux résultats 1 qui signifie qu'il Ya augmentation et 0 qui signifie le contraire.

— **ATCD :**
Le sigle ATCD a pour signification Antécédents. Ce sigle est classé dans les catégories : Sigles Médecine Composée de plusieurs résultats : HTA (augmentation anormale de la pression du sang, 0 qui signifie néant, embolie pulm, HTA et AVC, deux avortements, ASTHME, Cardiomyopathies, ULCERE Gastrique, Dyslipidémie, BASEDOW, TERRAIN ATOPIE etc. ...

— **DUREE HOSP :**
Correspond à la durée hospitalisation et s'exprime en jours.

— **DESIQUILIBRE :**
Qui regroupe la nature du niveau de taux de glycémie pour chaque patient soit :

- **Normoglycémie :** : qui signifie un taux de glycémie normal
- **Hyperglycémie :** : qui signifie une concentration en glucose dans le sang (glycémie) anormalement élevée.
- **Cetose :** : complication fréquente et aiguë du diabète
- **Acidocétose :** : qcaractériser par l'hyperglycémie, souvent supérieure à 20 mmol/L, avec la présence de corps cétoniques dans le sang ou l'urine.

— **Fièvre :**
qui est une réaction normale à une infection qui augmente la température du corps.Composée de deux résultats : 1 qui signifie la présence de la fièvre et 0 qui signifie néant.

— **TOUX :**
qui est une expulsion d'air des poumons subite et explosive.Composée de deux résultats : 1 qui signifie la présence de la toux et 0 qui signifie néant.

— **Anorexie :**
C'est un symptôme qui correspond à une perte de l'appétit. Composée de deux résultats : 1 qui signifie la présence d'anorexie et 0 qui signifie néant.

— **Agueusie :**
C'est un symptôme qui correspond à une perte de gout. Composée de deux résultats : 1 qui signifie la présence d'agueusie et 0 qui signifie néant.

— **ODYNOPHAGIE :**
est un symptôme qui se caractérise par une douleur lors de la déglutition. Composée de deux résultats : 1 qui signifie la présence d'odynophagie et 0 qui signifie néant.

— **RHINORRER**
est un symptôme qui se caractérise par la perte de l'odorat. Composée de deux résultats : 1 qui signifie la présence d'anosmie et 0 qui signifie néant.

— **Anosmie :**
est un écoulement de sécrétions, claires ou infectées, par le nez. Composée de deux résultats : 1 qui signifie la présence De RHINORRER et 0 qui signifie néant.

— **Courbature :**
Douleurs musculaires qui apparaissent suite à un effort physique intense ou inhabituel. Composée de deux résultats : 1 qui signifie la présence de courbature et 0 qui signifie néant.

— **Myalgie :**
est un symptôme qui se manifeste par des douleurs musculaires.Composée de deux résultats : 1 qui signifie la présence de myalgie et 0 qui signifie néant.

— **Frisson :**
est un symptôme qui se manifeste par un tremblement irrégulier, dû à la fièvre. Composée de deux résultats : 1 qui signifie la présence de frisson et 0 qui signifie néant.

— **Céphalée :**
symptôme qui se manifeste par de plainte douloureuse centrée sur la région crânienne. Composée de deux résultats : 1 qui signifie la présence de céphalée et 0 qui signifie néant.

— **Diarrhée :**
trouble digestif qui consiste en une émission de selles généralement liquides ou molles, dans une quantité et à une fréquence plus élevée que la normale. Composée de deux résultats : 1 qui signifie la présence de diarrhée et 0 qui signifie néant.

— **Insuffisance rénale :**
Composée de deux résultats : 0 qui signifie néant, 1 qui signifie qu'il insuffisance rénale.

— **Cytolyse :**
désigne la cytolyse hépatique représente la mort des cellules hépatiques.

— **Cytolyse :**
désigne la cytolyse hépatique représente la mort des cellules hépatiques.Composée de deux résultats : 1 qui signifie la présence de la cytolyse et 0 qui signifie néant.

— **Oxygène :**
détermine le taux d'oxygène consommé par patient.Composée de deux résultats : <9l et >9l. Corticoïdes : produits synthétiquement pour avoir les mêmes actions que le cortisol (ou cortisone), une hormone stéroïde produite par la couche externe (cortex) des surrénales. Composée de deux résultats : 1 qui signifie la présence de la corticoïde et 0 qui signifie néant.

— **Antalgiques :**
médicament qui atténue ou supprime la douleur sans en traiter la cause. Composée de deux résultats : 1 qui signifie la consommation d'antalgique et 0 qui signifie néant.

— **Antibiotiques :**
Composée de deux résultats : 1 qui signifie la consommation d'antibiotique et 0 qui signifie néant.

— **Macrolide :**
est un antibiotique généralement bactériostatique. Composée de deux résultats : 1 qui signifie la consommation de macrolide et 0 qui signifie néant.

— **Aminosides :**
sont des antibiotiques bactéricides utilisables en première intention par voie parentérale dans les infections sévères à germes Gram négatif aérobies. Composée de deux résultats : 1 qui signifie la consommation d'aminoside et 0 qui signifie néant.

— **Anticoagulant :**
médicament destiné à empêcher la formation de caillots sanguins. Composée de deux résultats : 1 qui signifie la consommation d'anticoagulant et 0 qui signifie néant.

— **VIT-ZINC :**
désigne la vitamine C et le zinc et ces deux participent au fonctionnement normal du système immunitaire. Composée de deux résultats : 1 qui signifie la consommation de vit-zinc et 0 qui signifie néant.

— **INSULINE :**
L'insuline est une hormone naturellement produite par le pancréas, plus précisément par des cellules spécialisées situées dans les îlots de Langerhans. Elle permet au glucose (sucre) d'entrer dans les cellules du corps. Composée de deux résultats : 1 qui signifie la consommation d'insuline et 0 qui signifie néant.

— **BIGUANIDE :**
représentent une des principales classes des antidiabétiques oraux. Composée de deux résultats : 1 qui signifie la consommation de Biguanide et 0 qui signifie néant.

— **Réanimation :**
Composée de deux résultats : 1 qui signifie le patient a été réanimé et 0 qui signifie néant.

— **Décès :**
Composée de deux valeurs : 1 signifie que le décès du patient a eu lieu pendant la période de suivi et 0 signifie que la donnée est censurée c'est-à-dire que l'évènement (le décès du patient n'a pas eu lieu pendant la période de survie.

— **Evolution :**
Composée de deux valeurs possibles : favorable et décès

La figure ci-dessous montre une partie de notre jeu de données :

	ADMISSION	SEXE	AGE	Tranche_Age	Test_PCR	ETAT__DIABETE	DIABETE	COOMORBI	HTA	ATCD	DUREE_HOSP	DESEQUILIBRE	Fievre___38__C
0	2020-06-05 00:00:00	F	65	ADULTE	FAUX NEGATIF	NON	0	1	1	HTA	9	NORMOGLYCEMIE	1
1	2020-06-09 00:00:00	M	76	Personne Agée	POSITIF	DIABETE INAUGL	1	0	0	0	8	HYPERGLYCEMIE	1
2	2020-06-09 00:00:00	M	39	ADULTE	POSITIF	DT2	1	0	0	0	6	NORMOGLYCEMIE	1
3	2020-06-09 00:00:00	F	67	ADULTE	POSITIF	DT2	1	1	1	HTA	5	HYPERGLYCEMIE	1
4	2020-06-09 00:00:00	M	50	ADULTE	FAUX NEGATIF	DT2	1	0	0	0	5	NORMOGLYCEMIE	0
5	2020-06-16 00:00:00	M	33	ADULTE	POSITIF	NON	0	0	0	0	4	NORMOGLYCEMIE	1
6	2020-06-16 00:00:00	M	49	ADULTE	POSITIF	DT2	1	0	0	0	4	HYPERGLYCEMIE	1
7	2020-06-18 00:00:00	M	92	Personne Agée	POSITIF	NON	0	0	0	0	8	NORMOGLYCEMIE	0
8	2020-06-18 00:00:00	F	69	ADULTE	POSITIF	DT2	1	0	0	0	4	NORMOGLYCEMIE	1

FIGURE 4.1 – Visualisation de notre dataset

4.1.1 Transformation des variables nécessaire pour la prédiction de survie

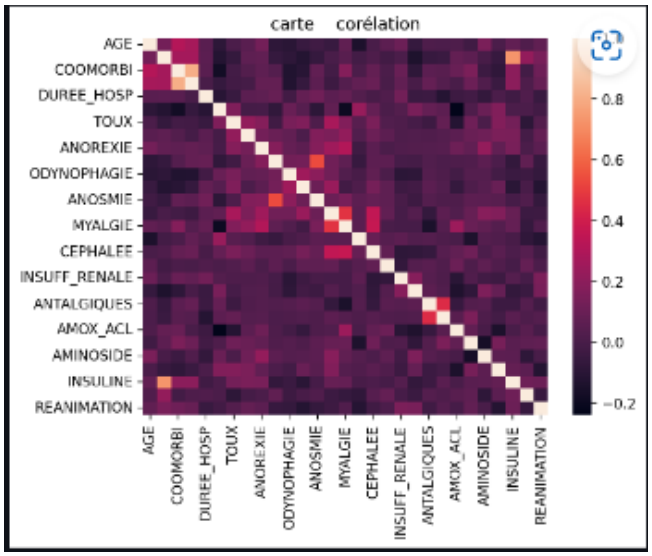


FIGURE 4.2 – Tableau de corrélation

- **Il existe des corrélations pour certaines de nos variables nous les avons réduit en une seule variable.**
- **Certaines de nos variables peuvent ne pas avoir une influence sur la variable cible.**
- **Certaines valeurs inconnues des variables nous avons procéder par des imputations.**
- **Pour les valeurs qui sont quasiment faibles et pour les autres nous les avons laissées comme telles.**

4.1.2 Encodage

Dans cette partie :

- **Pour nos variables contenant des valeurs non-colinéaires, nous avons décidé d'encoder ces valeurs par l'encodage OneHotEncoder.**

- **Exemple : Pour la variable déséquilibre**
  - **Avant encodage**

	Désiquilibre
0	NORMOGLYCEMIE
1	HYPERGLYCEMIE
2	CETOSE
3	ACIDOCETOSE

FIGURE 4.3 – EXEMPLE de variable non encoder

- **Pour nos variables contenant des valeurs colinéaires, nous avons décidé d'encoder ces valeurs par l'encodage OrdinalEncoder :**

	ACIDOCETOSE	CETOSE	HYPERGLYCEMIE	NORMOGLYEMIE
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0

FIGURE 4.4 – EXEMPLE d'encodage OneHotEncoder.

— **Exemple : Pour la variable sexe**
  — **Avant encodage**

	SEXE
0	F
1	M

FIGURE 4.5 – EXEMPLE de variable non encoder

— **Après encodage**

	SEXE
0	0
1	1

FIGURE 4.6 – EXEMPLE d'encodage ordinaleEncoder.

4.1.3 Equilibrage des données d'apprentissage

4.1.3.1 Problèmes de faible généralisation : l'Over-fitting et l'Under-fitting

Les problèmes d'une faible généralisation sont les causes principales des mauvaises performances des modèles prédictifs des algorithmes de machine Learning. Un modèle avec de bonnes performances est un modèle qui, après avoir appris des données d'apprentissage, fait des prédictions qui sont proches de la réalité si on lui présente de nouvelles données, c'est-à-dire si on le généralise.[3]

On parle de sur-apprentissage ou Over-fitting lorsque le modèle prédictif fait de très bons résultats sur les données d'apprentissage mais fera de mauvais résultat sur des données qu'il n'a pas encore vues. Cela est dû au fait que le modèle apprend les détails et le bruit dans les données d'apprentissage. Ainsi il ne pourra pas s'appliquer correctement aux nouvelles données, ce qui rendra mauvaises toutes les prédictions sur ces dernières.[5]

Dans le cas le Under-fitting ou sous-apprentissage, le modèle de prédiction ne parvient ni à faire une modélisation des données d'apprentissage ni à faire de bonnes prédictions sur les nouvelles données.

Ainsi retenons qu'il est essentiel de s'assurer que nos modèles prédictifs ne souffrent pas de problème de faible généralisation.
Dans cette partie :

— **Nous avons remarqué un déséquilibre au niveau des valeurs de la variable cible.**
— **Nous avons fait un sur échantillonnage pour équilibrer les valeurs de la variable cible .**

Date début hospitalisation

— **Résultat avant sur échantillonnage :**

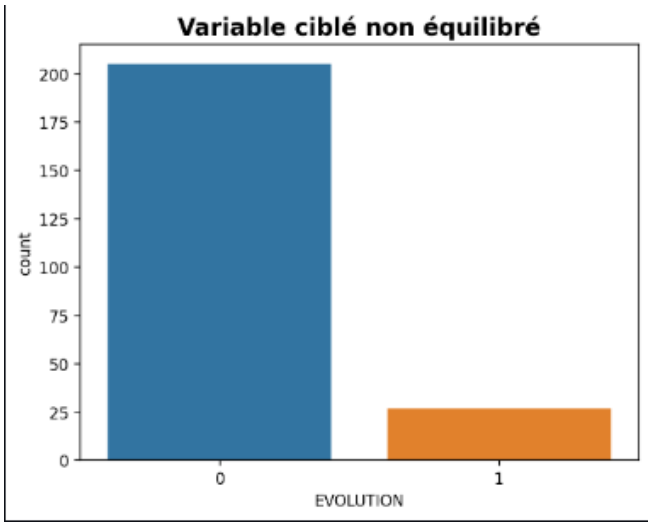


FIGURE 4.7 – Histogramme sur la répartition des données entre les deux classes de la variable cible

— **Résultat après sur échantillonnage :**

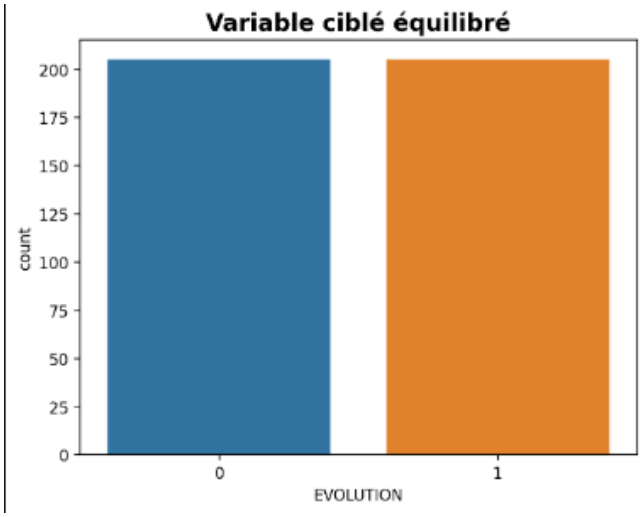


FIGURE 4.8 – Histogramme sur la répartition des données entre les deux classes de la variable cible aprés équilibrage

4.1.3.2 Conclusion

Avec cette répartition nous serons confrontés probablement à un problème de faible généralisation. En effet, vu le fort taux de cas de non COVID-19 par rapport au cas COVD-19, les futures prédictions du modèle risquent d'être en faveur de la classe majoritaire

(non COVD-19). Pour éviter de mauvaises performances du modèle prédictif, nous devrons équilibrer notre Dataset. Cependant l'équilibrage ne se fera que sur les données d'apprentissage. Ainsi nous allons diviser les données en deux parties : 70pourcent pour les données d'apprentissage et 30pourcent. Ainsi après avoir équilibré nos données d'apprentissage, nous disposons maintenant d'une nouvelle banque de données composé de 410 patients dont 205 de patients ayant subi de COVID-19 et 205 n'en ayant pas de COVID-19.

4.2 Evaluation générale des performances des modèles de classification

4.2.1 Les métriques de performance

Les métriques de performance permettent d'évaluer la performance de nos modèles. Leur utilisation dépend de plusieurs facteurs tels que le type de problème d'apprentissage, le contexte du problème à résoudre ainsi que le type de données. Ces métriques permettent aussi de faire une comparaison de la performance des différents modèles de Machine Learning sur les données afin de choisir celui qui offre une meilleure performance.
Pour les algorithmes de classifications de la méthode d'apprentissage supervisé, on étudiera principalement trois types de métriques de performance qui sont :

4.2.1.1 La matrice de confusion

LUne matrice de confusion est un tableau de $n * n$ dimension qui permet de visualiser les valeurs des modèles prédictifs en croisant les classes cibles réelles avec les classes cibles prédictives obtenues. Cette matrice nous donne le nombre d'instances correctement classées et le nombre d'instances mal classées.

		Classes actuelles	
		Positive	Négative
Classe prédictive	Positive	**VP**	**FP**
	Négative	**FN**	**VN**

FIGURE 4.9 – Exemple de matrice de confusion pour une classification binaire

- **VP :** c'est le nombre des vrais positifs, le nombre d'instances positives correctement classées
- **FP :**c'est le nombre des faux positifs, le nombre d'instances négatives prédites comme positives
- **FN :** c'est le nombre des faux négatifs, le nombre d'instances positives classées comme négatives
- **VN :**c'est le nombre des vrais négatifs, le nombre d'instances négatives correctement classées.

Ainsi à partir de ces informations, nous pouvons calculer plusieurs métriques qui sont :

- **Le taux de succès (Accuracy ou Hit Ration)**

L'accuracy est la proportion des instances qui sont correctement classées. Généralement, avec des jeux de données symétriques (valeurs faux positifs et des faux négatifs sont les mêmes), le meilleur des modèles est celui ayant le plus grand taux de succès.

$$Accuracy = \frac{VP +}{VP \ + \ VN \ + \ FP \ + FN}$$

— **Le taux d'erreur (E)**

Il représente l'erreur globale de la classification.

$E \ = \ 1 \ - \ Accuracy$

— **Précision ou valeur prédictive positive**

La précision est le rapport des instances positives correctement prédites au total de celles positives prévues

$$Accuracy = \frac{VP}{VP \ + \ FP}$$

— **Recall ou rappel**

Le Recall encore appelé sensitivité est le rapport des instances positives correctement prédites à toutes les instances positives de la classe réelle.

$$Accuracy = \frac{VP}{VP \ + \ FP}$$

— **F-Score**

Le F1-Score est la moyenne pondérée de la précision et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. Il prend des valeurs entre 0 et 1. Cette métrique est utilisée lorsqu'on cherche une balance entre la précision et le rappel. La précision s'avère être plus simple à comprendre, cependant, elle fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire. Lorsque la disposition des classes est inégale, l'utilisation du F-Score est généralement plus efficace

$$F \ - \ Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

4.2.1.2 La statistique de Cohen Kappa

La statistique de Cohen Kappa encore appelé K Cohen est une métrique qui peut être très utile, surtout, lorsqu'on a des données non équilibrées ou des cas de classification multiple. Le K permet de mesurer l'accord entre deux classifieur.

$$K \text{ ou } Kappa \ = \frac{P_0 - P_e}{1 - P_e}$$

où

P_0 représente l'Accuracy (probabilité d'accord observé)

P_e représente la probabilité d'un accord aléatoire

$$P_e = P_{oui} + P_{non}$$

$$avec \ P_{oui} \ = \frac{VP + FP}{VP + VN + FP + FN} \times \frac{VP + FN}{VP + VN + FP + FN}$$

$$et \ P_{non} = \frac{VP + FP}{VP + VN + FP + FN} \times \frac{VP + FN}{VP + VN + FP + FN}$$

Le K Cohen permet de comparer le taux de succès observé avec celui qui serait obtenu si on suppose que les classifications sont générées de façon aléatoire.

— **Interprétation :**

K	Interprétation
< 0	Désaccord
0 — 0.20	Accord très faible
0.21 — 0.40	Accord faible
0.41 — 0.60	Accord modéré
0.61 — 0.80	Accord fort
0.81 — 1	Accord presque parfait

FIGURE 4.10 – Interprétation selon la valeur de K Cohen

4.2.1.3 La courbe de ROC

Inventée lors de la seconde guerre mondiale, la courbe de ROC encore appelée fonction d'efficacité du récepteur, est une mesure qui permet de déterminer la performance d'un classificateur binaire en catégorisant les séparateurs en deux groupes différents selon une ou plusieurs caractéristiques de chaque élément. Cette mesure de performance est graphiquement représentée sous forme d'une courbe qui donne le taux de vrais positifs en fonction du taux de faux positifs. Un classifieur aléatoire est utilisé comme une ligne de base. Il tracera une droite allant de (0,0) à (1,1) .

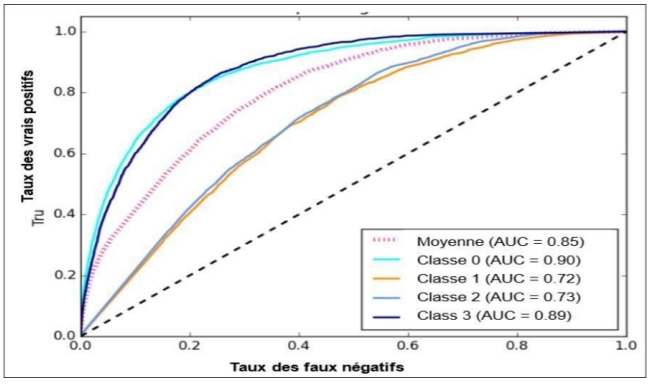


FIGURE 4.11 – Exemple de diagramme de ROC pour des classes multiples

Si l'aire sous la courbe (AUC) est égale à 1, le modèle fait une classification parfaitedes instances

— **Si AUC = 0.5, le modèle fait une classification au hasard.**
— **Si AUC < 1, le modèle fait une mauvaise classification des instances.**

4.2.2 Problèmes de faible généralisation : l'Over-fitting et l'Under-fitting

Les problèmes d'une faible généralisation sont les causes principales des mauvaises performances des modèles prédictifs des algorithmes de machine Learning. Un modèle avec de bonnes performances est un modèle qui, après avoir appris des données d'apprentissage, fait des prédictions qui sont proches de la réalité si on lui présente de nouvelles données, c'est -à-dire si on le généralise.

On parle de sur-apprentissage ou Over-fitting lorsque le modèle prédictif fait de très bons résultats sur les données d'apprentissage mais fera de mauvais résultat sur des données qu'il n'a pas encore vues. Cela est dû au fait que le modèle apprend les détails et le bruit dans les données d'apprentissage. Ainsi il ne pourra pas s'appliquer correctement aux nouvelles données, ce qui rendra mauvaises toutes les prédictions sur ces dernières.

Dans le cas le Under-fitting ou sous-apprentissage, le modèle de prédiction ne parvient ni à faire une modélisation des données d'apprentissage ni à faire de bonnes prédictions sur les nouvelles données.

Ainsi retenons qu'il est essentiel de s'assurer que nos modèles prédictifs ne souffrent pas de problème de faible généralisation.

4.2.2.1 Construction du modèle de prédiction

Pour construire notre modèle, nous avons utilisé python comme langage de programmation, jupyter comme éditeur de texte, et la bibliothèque libre de python destiné à l'apprentissage automatique, Scikit-Learn.
Ainsi nous allons décrire les différentes étapes pour la construction de notre modèle :

- **1ieme étape :** Tout d'abord nous avons importé toutes les bibliothèques auxquels on aura besoinpour la construction de notre modèle.
- **2ieme étape :** Ensuite, va venir de l'étape de nettoyage quand je dis nettoyage, je parle de la gestion des valeurs manquantes, la gestion des valeurs aberrantes, et aussi d'autres étapes de prétraitement de données.
- **3ieme étape :** Comme préparation de données pour la modélisation, il faut diviser ces algorithmes et en un ensemble de validation et de test.
- **4ieme étape :** Nous pouvons construire maintenant plusieurs model à partir de différents algorithmes de machine learning, il faut essayer plusieurs algorithmes faire des comparaisons et faire des choix.
- **5ieme étape :** Enfin nous allons finir par comparer la performance des algorithmes, sélectionner maintenant Le meilleur parmi eux et évaluer notre modèle final. - 6ieme étape : Sauvegarde du modèle dans notre disque et création de l'application web qui intègre le modèle. Dans ce dernier chapitre, il s'agira de développer une interface web permettant aux utilisateurs désirant se consulter de remplir leurs données et de voir le résultat.

Nous avons présenté plus haut plusieurs méthodes de classification. Parmi elles nous allons en comparer 5 qui sont : le Random Forest, la Regression logistique, le Bagging Classifier, le SVM et le GradientBoostingClassifier. Grace aux métriques de mesures vues précédemment, nous avons pu évaluer les performances des modèles crées avec ces 5 méthodes afin de retenir celle qui propose le meilleur modèle. Pour réaliser cette évaluation, 4 métriques ont été utilisées : l'accuracy, la précision, le Recall et le F1score.
Pour faire cette comparaison, nous utilisons la librairie python appelé Scikit Learn qui implémente ces méthodes d'apprentissage ainsi que les différentes métriques d'évaluation. Ainsi, nous récapitulons les résultats dans le tableau ci-dessous :

Méthodes	Précision	Recall	F1$_{S}core$	Accuracy
Random Forest	0.92	0.94	0.93	0.88
Regression logistique	0.96	0.75	0.84	0.75
SVM	0.96	0.72	0.830.18	0.72
GradientBoosting	0.10	100	0.18	0.10

FIGURE 4.12 – Récapitulatif des résultats suite à l'évaluation de 5 méthodes de classification

D'après ces résultats, on peut donc retenir que le Random Forest produit un meilleur score avec un taux d'Accuracy de 88 rcent de Précision de 94pourcents, de recall de 94pourcents et de F1-Score de 94pourcents. Il sera donc le modèle qu'on utilisera pour construire notre modèle.

Chapitre 5

Intègre du modèle de prédiction dans une application web

5.1 Analyse conceptuelle

5.1.1 Définition d'UML

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. UML est à présent un standard adopté par management Group(OMG).

5.1.2 Le processus unifié

L'UP utilise le langage UML. Beaucoup le considère comme une solution idéale pour remédier à l'éternel problème des développeurs. En effet, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en un système logiciel.

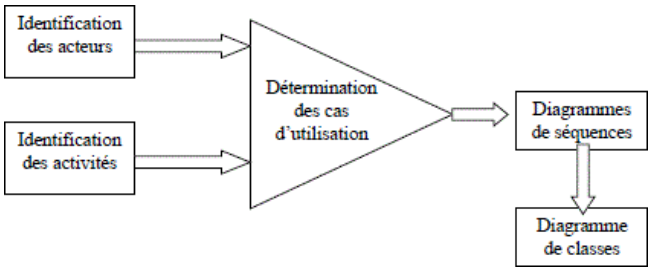


FIGURE 5.1 – : Les différentes étapes du processus unifié

5.1.3 Diagramme de cas d'utilisation

Pour modéliser en UML on utilise des diagrammes. Ces derniers sont répartis en deux groupes qui sont :

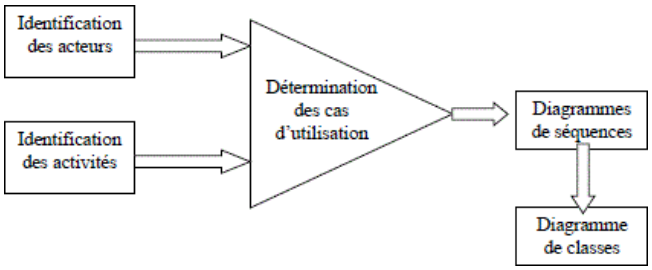


FIGURE 5.2 – : Les différentes étapes du processus unifié

- **Les diagrammes de structures :**
  - **Diagrammes de classe**
  - **Diagrammes d'objet**
  - **Diagrammes de composant**
  - **Diagrammes de déploiement**
  - **Diagrammes de structure composite**
  - **Diagrammes de package**
- **Les diagrammes de comportements :**
  - **Diagrammes d'activité**
  - **Diagrammes de cas d'utilisation**
  - **Diagrammes d'états**
  - **Diagrammes de séquence**
  - **Diagrammes communication**
  - **Diagrammes de vue**
  - **Diagrammes de timing**

Ainsi, nous présenterons un diagramme de cas d'utilisation qui permettra d'expliquer les fonctionnalités de notre application.

- **Diagramme de cas d'utilisation :**

  Encore appelé use cases, les diagrammes de cas d'utilisation sont utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

  Notre plateforme comporte un seul acteur : le médecin. En effet le médecin peut recueillir des informations sur l'état de COVID-19 à savoir les causes, les symptômes, qui contacter en cas pique d'AVC, quels sont les mesures à prendre le temps que les secoureurs ne viennent. Ensuite le médecin peut remplir une suite d'information via un formulaire. Ses données seront utilisées sur notre modèle d'apprentissage et nous permettra de dire au patient est – ce qu'il risque d'être décédé du COVID-19 ou pas.

  Ainsi notre diagramme de cas d'utilisation décrivant notre application est le suivant :

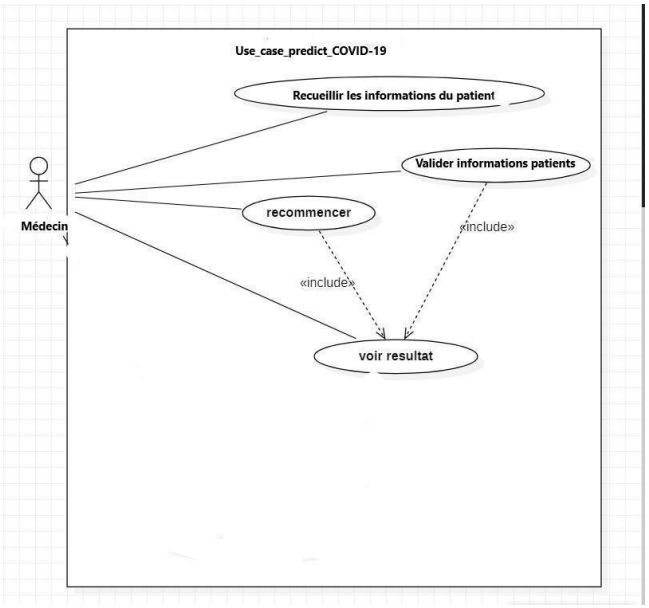


FIGURE 5.3 – Digramme de cas d'utilisation de notre application

5.2 Présentation des technologies utilisées

Pour le déploiement de notre application nous utiliserons principalement le Framework python Streamlit qui est un Framework open-source Python spécialement conçu pour les ingénieurs en machine learning et les Data scientists. Ce Framework permet de créer des applications web qui pourront intégrer aisément des modèles de machine learning et des outils de visualisation de données.

FIGURE 5.4 – Logo python et Streamlit

5.3 Présentation de l'interface

Nous avons une interface, possédant un formulaire de collecte de données du médecin et enfin un récap des infos saisies et notre résultat de prédiction.

1er affichage :

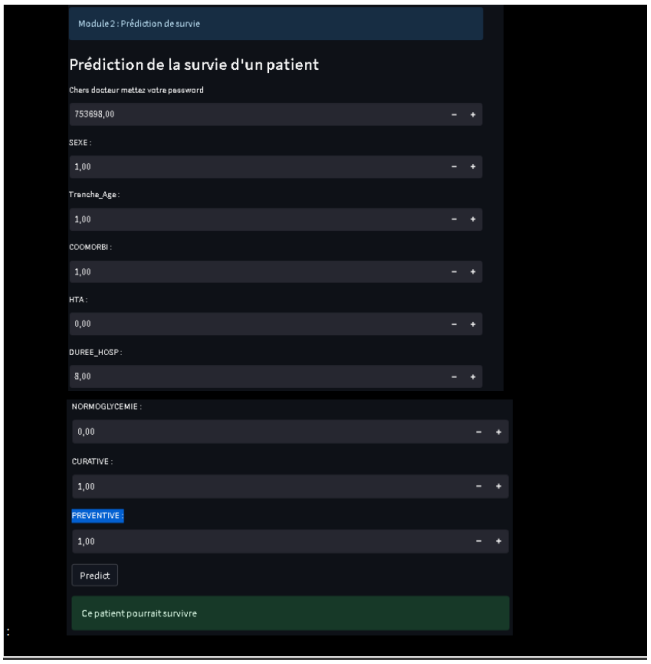


FIGURE 5.5 – Affichage du résultat

CONCLUSION

Dans ce travail de mémoire de Master nous avons utilisé nos connaissances pour réaliser : Une analyse de survie sur nos données sur divers groupes (sexe et tranche d'âge) afin de savoir la différence significative dans le taux de survie en générant dans chaque groupe selon la catégorie du groupe : la liste des probabilités de survie et tracer la courbe de survie.

Un apprentissage automatique ou Machine Learning pour créer un système de prédiction de survie de COVID-19. Après avoir construit et choisir le meilleur modèle de prédiction, nous l'avons intégré dans une application web développée avec le Framework Streamlit. Ainsi, tout un décideur (médecin) pourra utiliser pour prédire la survie d'un patient.

Pour la réalisation de ce projet nous étions confrontés à certaines difficultés qui sont :

Partie I :

- **Comprendre l'approche des domaines particuliers des statistiques qui est l'analyse de survie en particulier l'estimateur de survie de Kaplan Meier.**
- **La division de nos données en groupes afin de faire l'analyse sur divers groupes.**
- **Affichage de la liste des probabilités de survie qui listent les pourcentages de survie des patients par catégorie dans chaque groupe nous donnant les informations sur leurs pourcentages de survie.**
- **Affichage de la courbe de survie de Kaplan Meier.**

Partie II :

- **Comprendre l'approche mathématique des méthodes de classification.**
- **L'équilibrage de nos données**
- **Retraitement de nos données**
- **Déploiement sur notre application Streamlit**

Ce travail nous a permis de découvrir les domaines particuliers des statistiques qui sont l'analyse de survie en particulier, l'estimateur de survie de Kaplan Meier, les différentes fonctions de survie ainsi que leur principe de fonctionnement, l'apprentissage automatique, les familles d'apprentissage automatique, les différentes méthodes d'apprentissage automatique, ainsi que leur principe de fonctionnement.

En perspective. Il serait plus intéressant de donner au décideur (médecin) la possibilité, d'après les données saisies, de savoir quelles sont celles qui sont à l'origine du décès. Connaissant la cause, on peut lui proposer des conseils afin de l'éviter. Il serait aussi important de lui donner l'information du pourcentage de survie. Sur la plateforme, on pourrait aussi mettre un répertoire contenant les cordonnées de spécialistes à lui proposer pour une consultation en cas de nécessité.

ANNEXE

Présentations des différents algorithmes de Machine Learning étudié lors de nos tests

```
######import des bibliotheques######
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
##bibliotheque pour le decoupage des données d'apprentissage et de test
from sklearn.model_selection import train_test_split
##pour l'équlibrage
from imblearn.over_sampling import SMOTE,RandomOverSampler
##import pour la transformation des données en string
from sklearn import preprocessing
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

```

FIGURE 5.6 – Import des bibliothèques de Machine Learning

```
##Division des données en données train et données test 70% pour train , 30% pour test et 30% pour validation
# Partitionnement du jeu de données
from sklearn.model_selection import train_test_split
seed = 111
X = data.drop('EVOLUTION', axis=1)
y = data['EVOLUTION']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = seed, stratify= y)
## stratify= y  preserve la distribution de data['EVOLUTION']
```

FIGURE 5.7 – Retraitement des données et découpage en données de texte et d'apprentissage

```
### METHODE DE SUR-échantillonage
from sklearn.utils import resample
X2 = X_train
# ajout de la colone evolution dans X2
X2['EVOLUTION'] = y_train.values
X2.shape

minority = X2[X2.EVOLUTION == 1]
majority = X2[X2.EVOLUTION == 0]
#gclenfloge de la classe minoritaire pour quil vienne au meme niveau que la classe majoritaire
# n_samples recupere le nbre de ligne majority
minority_upsampled =resample(minority, replace =True , n_samples = len(majority))
```

FIGURE 5.8 – Equilibrage de données

```
#  Technique utilisé pour trouver les meilleurs hyperparametres pour nos données

## avec La LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression

logreg=LogisticRegression(random_state= seed, max_iter=500)
#hyp doit etre >0
logreg_hyp = {'C' : [0.001, 0.01, 0.1, 1, 10, 100]}

#cv signifie le nbre de plit quil fera
# exemple cv = 5 va diviser Les données d'entrainement en 5 plit a savoir
# ere interaction 4 plit comme données dentrainement et 1 plit comme données de test  apres
# il evaluer le modèle et chercher les parametre qui sont bon

logreg_cv = GridSearchCV(logreg, logreg_hyp, cv =5)

logreg_cv.fit(X_train_up, Y_train_up)
# Trouver le score et le meilleurs hyperparametre
print(logreg_cv.best_score_)
print(logreg_cv.best_estimator_)


##Avec le RandomForestClassifier
rf = RandomForestClassifier(random_state = seed)

#hyp doit etre >0
rf_hyp = {'n_estimators':[5, 10,20,50,100,200],
          'max_depth': [None, 2, 5, 10, 20] }

#cv signifie le nbre de plit quil fera
# exemple cv = 5 va diviser Les données d'entrainement en 5 plit a savoir
# ere interaction 4 plit comme données dentrainement et 1 plit comme données de test  apres
# il evaluer le modèle et chercher les parametre qui sont bon

rf_cv = GridSearchCV(rf, rf_hyp, cv =5)

# Trouver le score et le meilleurs hyperparametre
rf_cv.fit(X_train_up, Y_train_up)
print(rf_cv.best_score_)
print(rf_cv.best_estimator_)
## Avec le SVM
from sklearn.svm import SVC

sv = SVC(random_state = seed)

sv_hyp = {'C' : [0.1, 1, 10], 'kernel' :['linear', 'rbf']}

sv_cv = GridSearchCV(sv, sv_hyp, cv =5)

# Trouver le score et le meilleurs hyperparametre
sv_cv.fit(X_train_up, Y_train_up)
print(sv_cv.best_score_)
print(sv_cv.best_estimator_)


## Avec le BaggingClassifier
from sklearn.ensemble import BaggingClassifier
bc = BaggingClassifier(random_state = seed)

#hyp doit etre >0
bc_hyp = {'n_estimators':[5, 10,20,50,100,200]}

#cv signifie le nbre de plit quil fera
# exemple cv = 5 va diviser Les données d'entrainement en 5 plit a savoir
# ere interaction 4 plit comme données dentrainement et 1 plit comme données de test  apres
# il evaluer le modèle et chercher les parametre qui sont bon

bc_cv = GridSearchCV(bc, bc_hyp, cv =5)

# Trouver le score et le meilleurs hyperparametre
bc_cv.fit(X_train_up, Y_train_up)
print(bc_cv.best_score_)
print(bc_cv.best_estimator_)

##Avec le  GradientBoostingClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV
param_test1 = {'n_estimators':range(20,81,10)}
gsearch1 = GridSearchCV(estimator = GradientBoostingClassifier(learning_rate=0.1,
min_samples_split=500,min_samples_leaf=50,max_depth=8,max_features='sqrt',subsample=0.8,random_state=seed),
param_grid = param_test1, scoring='roc_auc',n_jobs=4, cv=5)
gsearch1.fit(X_train_up, Y_train_up)
print(gsearch1.best_score_)
print(gsearch1.best_estimator_)
```

FIGURE 5.9 – Choix des meilleurs hyper paramètres

•

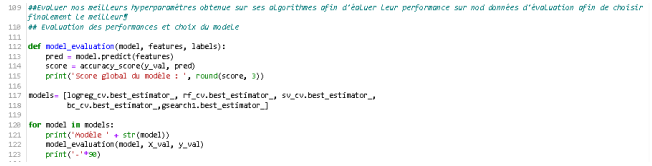


FIGURE 5.10 – Evaluation des performances et choix du modèle

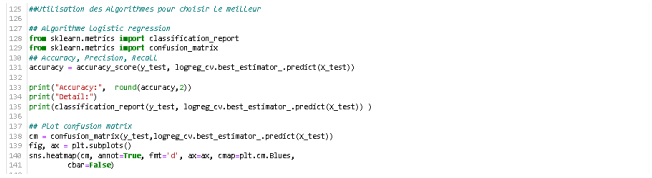


FIGURE 5.11 – Algorithme Logistic regression

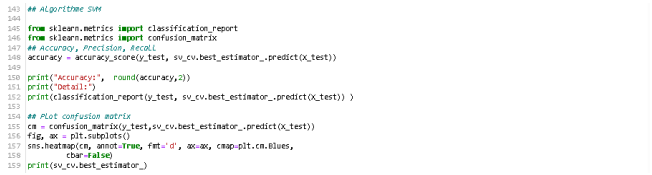


FIGURE 5.12 – Algorithme SVM

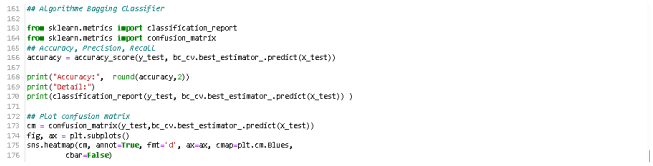


FIGURE 5.13 – Algorithme Bagging Classifier

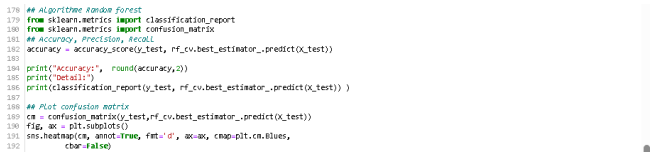


FIGURE 5.14 – Algorithme Random Forest Classifier

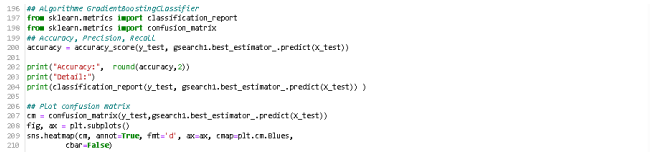
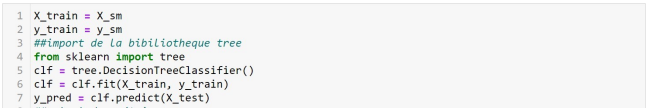


FIGURE 5.15 – Algorithme GradientBoostingClassifier

.



```
X_train = X_sm
y_train = y_sm
##import de la bibiliotheque tree
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

FIGURE 5.16 – Algorithme Algorithme d'arbre décisionnel

Bibliographie

[1] O. Aalen. *Non parametric estimation of partial transition probabilities in multiple decrement models.* Editeur1, 1978.

[2] Antoine Husson Anaël Beaugnon. *Le Machine Learning confronté aux contraintes opérationnelles des systèmes de détection.* Editeur1, 2017.

[3] Christopher M. Bishop. *Pattern recognition and Machine-Learning.* Editeur1, 2006.

[4] A. Necir Brahimi, D. Meraghni and L. Soltane. *Nelson-Aalen tail productlimit process and extreme value index estimation under random censorship.* Editeur1, 2018.

[5] Eyrollesn Cornuéjols, L. Miclet Y. Kodratoff. *Concepts et algorithmes.* Editeur1, 2002.

[6] Xavier Dupré. *Machine Learning, Statistiques et Programmation.* Editeur1, 2003.

[7] E.L Kaplan and P. Meier. *Non parametric estimation from incomplete observations. J. Amer. Statist. Assoc.* Editeur1, 1958.

[8] E. T Lee and J Wang. *Statistical methods for survival data analysis. John Wiley. J. Amer. Statist. Assoc.* Editeur1, (2003.

[9] Fabien Moutarde. *Apprentissage artificiel.* Editeur1, 1997.

[10] Míziou. *MÈmoire de master. Estimation non paramÈtrique, UniversitÈ Mohamed Khider, Biskra.J. Amer. Statist. Assoc.* Editeur1, (2014.

[11] M. Steinbach V. Kumar P. N. Tan. *MIntrodution to Data Mining.* Editeur1, 2006.

[12] Mc. Graw-Hill Science/Engineering/Math Thomas Mitchell. *Machine Learning.* Editeur1, 1997.