République Algérienne Démocratique et Populaire

Université A. MIRA de Béjaia

Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire présenté pour l'obtention du diplôme de master

Spécialité: Sciences de Données et Aide à la Décision

Étude prédictive sur la gestion du système de patients au CHUde Béjaïa. Étude de cas : mère et enfant.

Présenté par :
TABET Abdelhakim

Sous la direction de : Dr. L Asli

Défendu le 18/09/2024, devant le jury composé de :

Mr M. Soufit	M.C. classe/ A	Président de jury	UAMB - Bejaia.
Mr A.Laouar	M.C. classe/ A	Examinateur	UAMB - Bejaia
Mr S. Touati	M.C. classe/ A	Examinateur	UAMB - Bejaia.

Année Universitaire 2023 – 2024

Remerciments

Je tiens tout d'abord à exprimer ma gratitude envers **Allâh**, qui m'a accordé la patience et le courage nécessaires pour mener à bien ce travail.
Je tiens à exprimer mes profonds respects et remerciements à mon promoteur, **M. Larbi Asli**, dont la présence bienveillante, l'expertise inestimable et les conseils éclairés ont été les piliers essentiels de ma réussite tout au long de ce parcours.

Je remercie chaleureusement les membres du jury, **M Soufit** ,**M. Laouar** et **M Touati** , d'avoir accepté d'évaluer mon travail.

Je souhaite exprimer ma profonde gratitude envers le **personnel** dévoué du **Centre Hospitalier Universitaire de Béjaïa**. Leur engagement et leur dévouement envers les soins de santé sont exemplaires. Leur expertise, compassion et professionnalisme ont été d'une valeur inestimable pour mon projet. Je suis reconnaissant envers chaque membre du personnel qui a généreusement partagé son temps, ses connaissances et son expérience avec moi.

Je souhaite également exprimer mes sincères remerciements à **ma famille** qui m'a soutenu tout au long de cette aventure. Leur encouragement constant, leur soutien moral et affectif ont été d'une importance cruciale pour moi. Leur patience et leur compréhension inconditionnelle ont été essentielles dans la réalisation de ce travail. Je leur suis profondément reconnaissant pour cet amour indéfectible.

Enfin, je tiens à remercier **toutes les personnes** qui ont contribué de près ou de loin à l'élaboration de ce mémoire. Votre soutien et votre contribution ont été d'une importance capitale, et je vous en suis profondément reconnaissant.

Dédicaces

Quelles que soient mes paroles et mes actions, je demeure incapable de vous témoigner une reconnaissance à la hauteur de vos mérites. Les mots les plus choisis ne sauraient véritablement exprimer l'ampleur de ma gratitude et de ma reconnaissance. C'est avec une profonde dévotion que je vous dédie ce travail, témoignage de vos efforts inlassables et le reflet de mon amour sincère.

À mon très cher Père

Gratitude infinie pour ton amour inconditionnel, tes conseils avisés, ma plus grande inspiration. Ce mémoire est le fruit de notre complicité et de ton soutien inestimable.

À ma très chère mère

Tes sacrifices et ta dévotion sont incommensurables, ton amour inébranlable et ta force infinie m'ont porté tout au long de ce parcours. Ce travail est un témoignage de ma profonde gratitude envers toi, tu es la source de ma détermination et de ma réussite, merci d'avoir été mon roc, ma confidente et ma plus grande admiratrice.

À mon frère Razik qui passe également son baccalauréat cette année, j'espère qu'il réussira brillamment.à ma petite sœur Rbiha, à ma tante, ma cousine Nassila, et à tous les membres de ma famille , merci d'avoir toujours été là pour moi, de m'écouter, de me soutenir et de partager des moments précieux. Notre lien fraternel est une bénédiction que je chéris profondément.

À ma précieuse famille Anis qui se trouve à l'étranger,À Sidou, présent à mes côtés depuis le début jusqu'à la fin de ce mémoire, et à Hassan pour son soutien précieux .Votre amitié sincère et votre soutien inébranlable ont été une source de bonheur et de réconfort pour moi.

Contents

List of Figures

List of Tables

Liste des abriviations

CHU Centre Hospitalier Universitaire

S/D Sous-Directions

AM Activités Médicales

AP Activités Paramédicales

GADM Gestion Administrative Du Malade

FD Formation et Documentation

P Personnel

BC Budget Compatibilité

CC Calcul Des Coûts

SE Services économiques

EIF Equipements, Infrastructure et Maintenance

PF Produits Pharmaceutiques

ML Machine Learning

RF Random Forest

Introduction générale

Le secteur de la santé est en constante évolution, confronté à des défis complexes tels que le vieillissement de la population, l'augmentation des maladies chroniques et la nécessité d'optimiser les ressources disponibles. Avec l'avènement des technologies de l'information et l'accroissement des données disponibles, l'analyse de ces données est devenue cruciale pour améliorer la qualité des soins, optimiser les processus hospitaliers et soutenir les décisions cliniques. Dans ce contexte, la capacité à utiliser des données de santé pour identifier des solutions innovantes représente une opportunité majeure pour les professionnels de la santé et les chercheurs. C'est dans cette optique que on a choisi de réaliser un stage au Centre Hospitalier Universitaire (CHU), où on pourrai appliquer nos compétences en analyse de données pour contribuer à l'amélioration de la gestion des soins aux patients.

Une profonde motivation pour travailler avec des données réelles, combinée à la mise en pratique de mes connaissances dans mon domaine d'expertise. Un intérêt particulier pour le secteur de la santé, où l'opportunité unique de traiter des données médicales permet de proposer des solutions concrètes face aux défis complexes de ce domaine. Le dynamisme du secteur, où l'analyse des données améliore la qualité des soins, optimise les processus hospitaliers et favorise des décisions cliniques plus éclairées et efficaces, a fait du stage au CHU un choix évident..

Le Centre Hospitalier Universitaire de Béjaïa, l'un des principaux établissements de santé de la région, regroupe cinq centres hospitaliers. Cette étude se concentrera sur l'unité mère-enfant de Targa Ouzemour.

Un autre défi majeur réside dans l'absence d'une gestion efficace et claire des données médicales des patients. Actuellement, les agents administratifs ne parviennent pas à exploiter ces informations de manière optimale pour assurer une gestion efficiente. De plus, en raison de la mauvaise gestion des données à l'échelle du CHU de Béjaïa, les employés n'ont pas une vision précise des patients présents ni des admissions à venir, ce qui les empêche de gérer efficacement les futurs flux de patients à la maternité.

Conception d'un modèle prédictif pour répondre aux besoins du bureau des admissions, avec pour objectif principal la prévision de la durée de séjour des patients. En tant que data scientist, le développement d'un modèle de machine learning basé sur l'analyse des informations disponibles au bureau des admissions permettra d'anticiper plus efficacement la durée de séjour des patients.

Pour prédire la durée de séjour, plusieurs étapes ont été suivies. Tout d'abord, il a été

nécessaire de bien comprendre et définir le problème. Ensuite, la centralisation et la collecte des données ont été effectuées. Les données ont été préparées afin d'optimiser l'efficacité de l'algorithme. Après le prétraitement, une analyse et une modélisation des données ont permis de valider les hypothèses. Une fois le modèle validé, il a été mis en production avec une visualisation simplifiée. Enfin, la qualité du travail a été assurée par la comparaison des performances avec d'autres algorithmes.

Ce mémoire s'appuie sur une expérience de stage réalisée au CHU de Béjaïa, où j'ai eu l'opportunité de travailler aux côtés d'une équipe pluridisciplinaire, composée du chef de service et des agents administratifs. Ce stage m'a permis de comprendre les enjeux spécifiques du CHU et d'acquérir une vision globale des besoins et des contraintes liées à la mise en place d'un modèle de machine learning.
Ce mémoire est organisé en quatre chapitres :

- **Le premier chapitre** présente l'organisme d'accueil, le CHU de Béjaïa, et inclut une étude préliminaire ainsi qu'une analyse au niveau du bureau des entrées.
- **Le deuxième chapitre** aborde quelques définitions sur la science des données et le machine learning, ainsi que leur relation avec le domaine de la santé.
- **Le troisième chapitre** Détail de la réalisation de l'objectif principal : la prédiction de la durée de séjour des patients, incluant les étapes suivies telles que l'analyse des données et l'application de techniques de machine learning.
- **Le quatrième chapitre** présente des critiques et des suggestions pour améliorer l'état du bureau des entrées, ainsi que les outils que j'ai utilisés au cours de ce mémoire.

Notre projet s'achèvera par une conclusion générale.

1

Contexte et cadre théorique

Introduction

Dans le domaine en perpétuelle évolution de la santé, l'intégration de l'informatique est devenue indispensable. La combinaison de la science des données et de ses techniques joue un rôle crucial dans l'amélioration des soins de santé. L'exploitation des vastes quantités de données disponibles dans le secteur médical permet d'identifier des tendances significatives et d'optimiser les processus cliniques et administratifs.

En utilisant la science des données et ses techniques avancées, il est possible de maximiser l'efficacité des ressources et d'améliorer la qualité des soins prodigués aux patients. Cette convergence entre l'informatique et la science des données ouvre de nouvelles perspectives pour relever les défis complexes du domaine de la santé, et offre des solutions innovantes pour répondre aux besoins croissants des patients et des professionnels de la santé.

1.1 Présentation du CHU (Centre Hospitalier Universitaire)

Le centre hospitalo-universitaire est un établissement public à caractère administratif, doté de la personnalité morale et jouissant de l'autonomie financière.

1.1.1 Création:

Le CHU de Béjaïa a été créé par le décret exécutif n° 09-319 du 17 Chaoual 1430 (6 octobre 2009), complétant la liste des centres hospitalo-universitaires annexée au décret exécutif n° 97-467 du 2 Chaabane 1418 (23 décembre 1997), lequel fixe les règles de création, d'organisation et de fonctionnement des centres hospitalo-universitaires.[7]

Figure 1.1: Le logo de CHUB

L'étude préliminaire porte sur le système d'information hospitalier du Centre Hospitalier Universitaire de Béjaïa, l'un des principaux établissements de santé de la région. Le CHU de Béjaïa est composé de cinq centres hospitaliers, mais cette étude se concentrera sur trois d'entre eux : Frantz Fanon, Khellil Amrane, et Targa Ouzemour. Voir l'organigramme à la figure1.2.

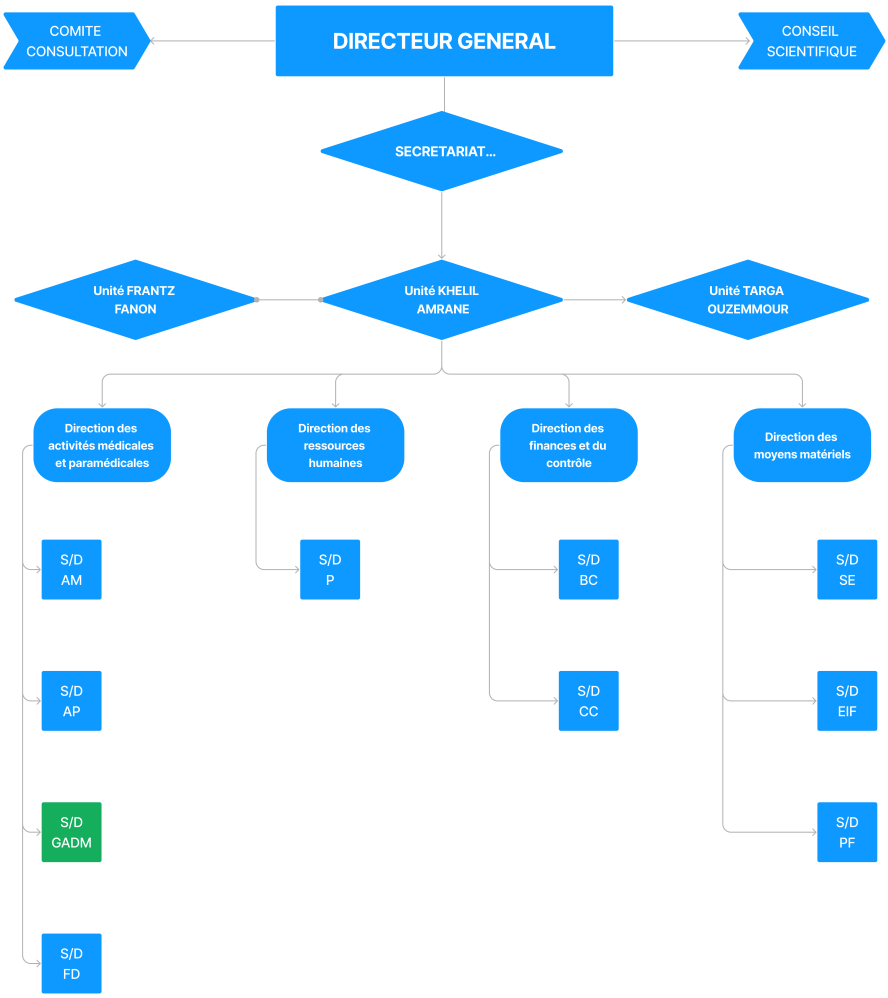


Figure 1.2: Organigramme du CHU de Bejaia

1.2 Unité mère et enfant

L'hôpital "Mère et Enfant" de Targa-Ouzemmour à Béjaïa a ouvert ses portes en 1964, succédant à "Le Beau Séjour", situé rue Fatima et auparavant désigné sous le nom d'ESH. Cet établissement se consacre principalement à la santé des femmes, couvrant divers aspects de la médecine féminine tels que la procréation, l'accouchement et les soins gynécologiques. Bien que le service de procréation et d'accouchement fonctionne 24 heures sur 24, le personnel médical et administratif demeure sur place pour garantir une prise en charge continue, témoignant de leur dévouement constant.[8].

Pour répondre aux besoins essentiels en matière de santé maternelle et infantile, l'hôpital a mis en place un service dédié, conforme aux directives de la direction générale des admissions. Un service intermédiaire, au bureau des admissions, assure la gestion quotidienne des dossiers des patientes, qu'elles soient mariées, célibataires ou veuves, ainsi que des enfants, depuis la conception jusqu'à l'âge de 28 jours.

La remise du bulletin de patient revêt une importance cruciale pour les familles, favorisant les rapprochements et les célébrations familiales, notamment lors des naissances. Le traitement des demandes est assuré quotidiennement, y compris les jours fériés et les jours de fête. En reconnaissance de leur travail, la Journée Internationale de la Sage-Femme est célébrée chaque année le 5 mars.

L'organigramme de l'annexe «Mère Et enfant» est présenté dans la figure 1.3.

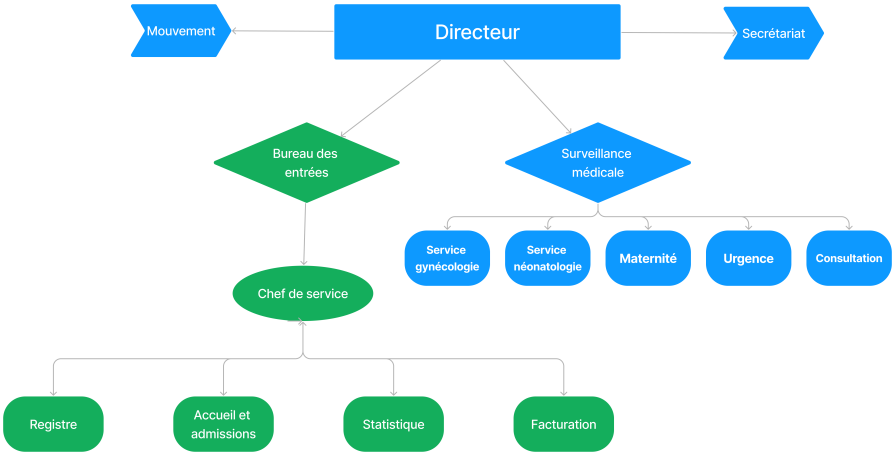


Figure 1.3: Organigramme de l'annexe «Mère Et enfant»

1.3 Stage

Pour la réalisation du projet de fin d'études, un stage a été effectué au CHU de Béjaïa. Lors de la première visite au bureau d'accueil,Un agent nous a directement orientés vers la direction des moyens et matériels. C'est là qu on a rencontré M. Ramy, un ingénieur en informatique. Après lui avoir exposé l'objectif du projet, une proposition a été faite d'effectuer une optimisation du système de gestion des patients au CHU de Béjaïa. Cette idée, jugée intéressante, a été retenue comme thème du mémoire. À l'issue d'une discussion fructueuse, le sujet du mémoire a été finalisé : "Étude prédictive sur la gestion du système de patients au CHUde Béjaïa. Étude de cas : mère et enfant."

Concernant le service d'accueil, également appelé *bureau des entrées*, il est composé de plusieurs agents, chacun ayant des tâches spécifiques. Deux bureaux, équipés chacun d'un PC, sont dédiés à l'inscription des patients en utilisant le logiciel PATIENT. Un autre agent est chargé de l'encaissement des frais d'accouchement, fixés à 100 DA par nuit. Enfin, certains agents sont responsables de la gestion et du remplissage des documents nécessaires.
Après avoir complété les formalités administratives nécessaires avec les confirmations de l'université et de l'hôpital, le deuxième jour de stage, **M. Ramy** a dirigé le stagiaire directement vers le bureau des entrées de l'unité mère-enfant à Targa Ouzmour. Lors de cette première visite à la

maternité de Targa Ouzmour, au bureau des entrées, une rencontre avec **Mme Zaidi**, la chef de service, a eu lieu. Elle a accueilli le stagiaire chaleureusement et a exprimé sa satisfaction de voir un stagiaire motivé et sérieux. Mme Zaidi a assuré son soutien maximal pour le projet en cours.

L'opportunité de travailler avec le logiciel de gestion des patients a été saisie, avec l'assistance de certains agents. Après avoir compris le fonctionnement du logiciel, une utilisation autonome a été possible. Lors de la saisie des informations dans le logiciel, une vérification avec l'accompagnant du patient (généralement un proche) a été effectuée pour garantir l'exactitude des données, telles que le nom, le prénom, la date de naissance, l'âge, les coordonnées de l'accompagnant, et les informations de contact. Une fois les informations validées, le logiciel a automatiquement imprimé le bulletin d'admission et le résumé standard de sortie, tout en attribuant un code unique d'identification. Par la suite, une fiche navette a été remplie manuellement pour accompagner le patient tout au long de son séjour à l'hôpital.

Après l'utilisation du logiciel **PATIENT**, j'ai constaté qu'il offre une base de données précieuse pour atteindre l'objectif fixé. L'opportunité de procéder à la déclaration d'une naissance a été saisie. Le processus a commencé avec le formulaire intitulé "Déclaration de Naissance", émis depuis la salle d'accouchement et contenant toutes les informations essentielles, telles que la date et l'heure de la naissance. Ensuite, en utilisant le logiciel PATIENT, la procédure a été suivie en sélectionnant d'abord le code correspondant à la naissance (numéro 5) et en remplissant toutes les informations requises. Après cette étape, le registre des naissances a été complété avant de procéder à la déclaration officielle auprès de l'Administration de la Population et des Citoyens (APC). Au cours de cette journée, Mme Zaidi a expliqué le processus de gestion des informations, de l'arrivée du patient jusqu'à sa sortie. Tout au long du stage, assiduité et désir constant de compréhension et d'enrichissement des connaissances ont été manifestés. À l'approche de la fin de cette expérience, gratitude sincère exprimée envers toutes les personnes ayant apporté une aide précieuse, de la cheffe de service jusqu'au dernier employé.

1.4 Description du fonctionnement du service d'accueil

1.4.1 flux d'information

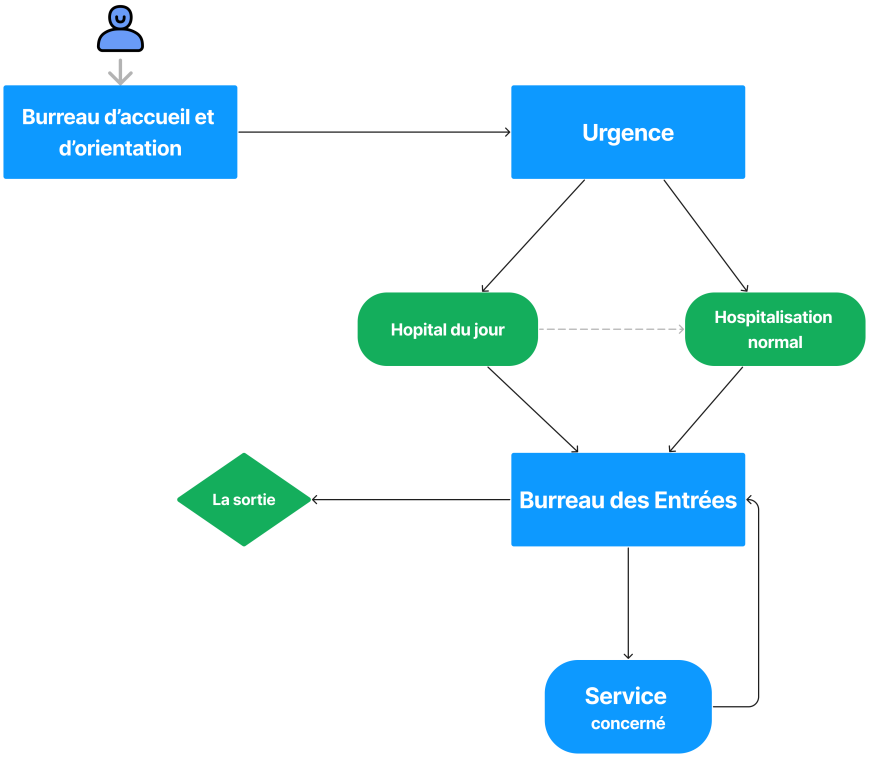


Figure 1.4: Flux d'information

Interprétation

À l'arrivée du patient à l'unité, il se présente d'abord au bureau d'accueil et d'orientation, également appelé bureau de tri, où son cas est évalué en fonction de son âge et de son sexe. Ensuite, il est dirigé directement vers le service des urgences. Après l'examen médical, les médecins décident si le patient doit être admis en hospitalisation de jour, en hospitalisation normale, ou s'il peut sortir directement. L'admission se fait alors au bureau des entrées.

En cas d'hospitalisation de jour après la consultation (dont la durée peut varier entre 4, 7, 8 ou 9 heures), le médecin décidera si une hospitalisation normale est nécessaire ou non. Si l'hospitalisation normale n'est pas requise, le patient devra se présenter au bureau des quittances pour payer sa facturation. En cas d'accouchement, le bureau des entrées procédera à l'admission et remettra une fiche de navette, un certificat de séjour, et un certificat de présence. Pour une hospitalisation normale, le patient devra également remplir une fiche de navette et demander un certificat de séjour et un certificat de présence au bureau des entrées.

Après la période d'accouchement, à la sortie, le patient doit revenir au bureau des entrées avec la fiche de navette remplie par les médecins et les infirmiers. Avant de quitter l'hôpital, le patient devra régler sa quittance (au tarif de 100 DA par jour). Le bureau des entrées complétera

les formalités nécessaires pour délivrer un billet de sortie au patient. La fiche de navette sera conservée par le bureau des entrées pour des études statistiques et archivage, et sera archivée après 5 ans.

1.4.2 Les registres

- **Registre des naissances:** Un registre consigne les informations sur les nouveau-nés, y compris leur nom, leur date de naissance, le nom des parents, ainsi que d'autres détails pertinents.
- **Registre des décès:** Un registre spécifique est utilisé pour enregistrer les décès survenus parmi les patients traités dans l'unité mère et enfant de l'hôpital. Il contient des informations telles que le nom du patient décédé, la date et l'heure du décès, la cause du décès, ainsi que d'autres détails pertinents.
- **Registre de la morgue:** Un registre documente les informations sur les corps des personnes décédées, spécifiquement ceux qui sont placés dans la morgue de l'unité mère et enfant de l'hôpital. Il comprend des détails d'identification du défunt, les circonstances du décès, ainsi que les autorisations et les demandes des familles. etc.
- **Registre minute:** Un registre ou un document est utilisé pour enregistrer les détails des réunions, des décisions ou des événements importants dans l'unité mère et enfant de l'hôpital. Il peut inclure des informations sur les discussions concernant les soins aux mères et aux enfants, les décisions médicales, ainsi que les actions entreprises.
- **Registre Répertoire :** Un registre répertorie les numéros d'identification uniques attribués aux patients, aux mères et aux enfants traités dans l'unité. Ces numéros sont utilisés pour suivre les dossiers médicaux, les traitements, les rendez-vous et d'autres activités liées aux soins maternels et infantiles.
- **Registre matricule :** Un registre répertorie les numéros d'identification uniques attribués aux patients, aux mères et aux enfants traités dans l'unité. Ces numéros sont utilisés pour suivre les dossiers médicaux, les traitements, les rendez-vous et d'autres activités liées aux soins maternels et infantiles.
- **Registre matricule hopital du jour :** Un registre utilisé pour enregistrer les patients et les enfants admis à l'hôpital de jour de l'unité mère et enfant. Il contient des informations sur les consultations, les traitements administrés, et les suivis médicaux pour les mères et les enfants traités en ambulatoire.

1.4.3 Les documents

- **Fiche navette:** Document utilisé pour suivre le patient depuis son admission jusqu'à sa sortie, contenant des informations sur les actes médicaux et paramédicaux réalisés, destiné à être consulté par le médecin et l'infirmier.

- **Fiche navette hopital du jour:** La fiche navette de l'hôpital de jour est un formulaire utilisé pour enregistrer la mise en observation des patients atteints de maladies. Actes médicaux pratiques dans l'etablissemnt d'hospitalisation y compris les consultation effectuées par les particiens externes au service.
- **Bulletin d'admission:** Le bulletin d'admission/sortie est émis lorsque le patient est admis au bureau des entrées.
- **Résumer clinique standard :** Le document accompagnant la fiche navette à l'admission, permettant au service d'enregistrement de prendre connaissance des antécédents médicaux, des traitements en cours et des informations pertinentes sur l'état de santé du patient pour assurer une prise en charge appropriée.
- **Certificat de séjour:** Document justifiant les séjours des patients, enregistrant la date d'entrée à l'hôpital.
- **Certificat de présence:** Document attestant de la présence des patients à l'hôpital.
- **La quittance:** Document justifiant le paiement effectué à la sortie de l'établissement.
- **Quittancier de facturation:** Document récapitulatif des transactions financières, indiquant à la fois les exonérations et les non-exonérations des frais.

1.4.4 Patient

Présentation de logiciel

Le logiciel Patient, voir son interface sur la figure 1.5, a été développé dans les années 90 par Madame ABDI LDJOUHER, chef du service informatique du CHU de Mustapha Bacha d'Alger, en collaboration avec Monsieur Benkaci du Ministère de la Santé, de la Population et de la Réforme Hospitalière. Pour sa conception, ils ont utilisé l'outil de programmation FOXPRO, désormais connu sous le nom de MS-DOS suite à son acquisition par Microsoft [1].

La première utilisation du logiciel a débuté le 1er janvier 1995, mais était initialement limitée au CHU de Mustapha Bacha d'Alger. Ce n'est qu'en 2001 que le logiciel Patient a été étendu à la plupart des établissements de santé algériens, permettant ainsi une gestion informatisée efficace des patients à travers tout le pays [1].

Figure 1.5: Interface de logiciel Patient

Description

Le logiciel Patient est une solution informatique intégrée développée pour gérer l'administration des patients dans les établissements de santé publics en Algérie. Ce logiciel comprend plusieurs programmes interconnectés, notamment pour les admissions, les transferts inter-services, les renseignements et les éditions, afin d'assurer une gestion efficace des entrées et sorties des patients. Au bureau des admissions de l'hôpital, Patient est utilisé pour enregistrer les informations relatives à l'admission des patients, telles que leur nom, prénom, âge, sexe, service attribué, nom du praticien responsable, ainsi que les coordonnées d'un membre de la famille ou d'un ami les accompagnant.

Fonctionnant sur le réseau LAN de l'hôpital, le logiciel Patient permet un accès rapide aux informations déjà enregistrées, facilitant ainsi la gestion efficace des patients. Il offre également la possibilité de modifier ou d'imprimer des documents tels que le résumé standard de sortie, le bulletin d'admission du patient et le bulletin du garde-malade. Les paramètres de configuration du logiciel, tels que le code d'identification de l'établissement, les listes des spécialités et des services proposés, les services par genre, ainsi que la nomenclature des médicaments, des maladies et des personnes exonérées ou non, sont personnalisables pour répondre aux besoins spécifiques de chaque établissement de santé. Les prix forfaitaires des actes médicaux peuvent également être ajustés selon les exigences locales[14].

Le logiciel Patient offre plusieurs fonctionnalités essentielles, notamment:

- Sauvegarder les informations d'un patient.
- Sauvegarder l'historique de mobilité.
- Sauvegarder les méta-données d'une admission (service d'affectation, la date de sortie et le praticien qui a demandé l'admission).
- Impression du résumé standard, les bulletins d'admission ou certificat de séjour.

- Récupérer les statistiques par plage de temps.

Comme tout logiciel utilisé dans les bureaux, le logiciel Patients présente à la fois des points positifs et des points négatifs. Ce tableau suivant 1.1 met en évidence les points positifs et les points négatifs spécifiques du logiciel Patient.

Avantages	**Inconvénients**
Les utilisateurs sont habitués à ses éventuels problèmes en raison de son utilisation à l'échelle nationale.	Aspect visuel peu attrayant, ce qui peut affecter l'expérience utilisateur.
Il est toujours opérationnel.	Accessible uniquement depuis le bureau d'accueil, ce qui limite l'accessibilité pour d'autres utilisateurs ou services.
Ne nécessite pas beaucoup de ressources.	Manipulation uniquement possible via le clavier, ce qui peut être moins intuitif pour certains utilisateurs.
Auto-complétion des champs de saisie comme les praticiens, les services ainsi que la date.	Les champs de saisie (input) sont invisibles, ce qui peut entraîner des erreurs de saisie ou de navigation.
	Ne sauvegarde que les métadonnées (nom, prénom, service, etc.), sans inclure d'informations médicales détaillées, ce qui limite l'utilité du système pour les professionnels de santé.
	Un patient déjà enregistré perd son identifiant pour la prochaine visite, ce qui peut entraîner une perte de suivi ou de continuité des soins.
	L'heure actuelle doit être saisie manuellement, ce qui peut être fastidieux et sujet à des erreurs.

Table 1.1: Les avantages et les inconvénients du logiciel Patient

1.5 Problématique

En travaillant au bureau des entrées de la maternité de Targa Ouzmour, j'ai été confronté à un flux constant de données comprenant des informations essentielles sur les patients, les services, les salles d'accouchement, et autres aspects cliniques. Malgré la richesse de ces données, leur gestion actuelle est inefficace, laissant place à des lacunes et à une vision fragmentée de l'ensemble des informations disponibles. Cette situation, où même les agents ne disposent pas d'une vue d'ensemble des données, crée un environnement de travail désorganisé.

Dans ce contexte, l'objectif du mémoire est l'identification et la proposition de solutions concrètes pour améliorer la gestion des données au sein des établissements de santé, avec un focus particulier sur la maternité de Targa Ouzmour. Les sections suivantes présenteront les objectifs spécifiques de l'étude ainsi que les stratégies et outils prévus pour atteindre une gestion optimale des données dans cet environnement médical crucial.

1.6 Objectif principal

L'objectif principal réside dans la proposition d'un plan complet visant à résoudre les problèmes de gestion des données identifiés et à améliorer globalement le système. À cet effet, les actions suivantes sont envisagées :

- Réaliser une visualisation approfondie et détaillée des données du système à l'aide d'outils informatiques avancés tels que le langage Python et des bibliothèques spécialisées comme Matplotlib et Seaborn. Cette visualisation permettra une compréhension holistique des flux de données, des tendances et des patterns émergents, facilitant ainsi une gestion plus éclairée et proactive des données.
- Utiliser des techniques d'estimation et de prédiction avancées, telles que les modèles de Machine Learning, pour anticiper les durées de séjour des patients. Ces prédictions aideront le personnel médical et administratif à planifier de manière proactive les ressources, à optimiser les processus de traitement et à fournir des soins plus efficaces et personnalisés aux patients.

En combinant ces stratégies, notre objectif est de créer un environnement de gestion des données robuste et orienté vers les résultats au sein de la Maternité de Targa Ouzmour, conduisant ainsi à une amélioration significative de la qualité des soins et de l'efficacité opérationnelle.

Conclusion

Ce chapitre inclut une description de l'expérience de stage au CHU de Béjaïa. L'observation approfondie des insuffisances et des difficultés dans la gestion manuelle des dossiers papier au sein de l'établissement met en lumière la nécessité d'un système de visualisation des données au bureau des entrées. L'importance de cette mise en place repose sur l'optimisation basée sur des données réelles. Cette approche est perçue comme un moyen d'améliorer la manipulation des données, la gestion des admissions et des sorties, tout en contribuant à une efficacité opérationnelle accrue et à une satisfaction des patients améliorée.

2

Fondements Théoriques de la Science des Données et du Machine Learning dans le Domaine de la Santé

Introduction

Nous commencerons par revisiter les bases du machine learning, en mettant en évidence ses avantages et ses défis, ainsi que son implémentation pratique à travers des langages de programmation comme Python. En outre, nous aborderons le processus d'estimation dans le contexte du machine learning, en soulignant l'importance de choisir des modèles robustes et fiables pour résoudre des problèmes complexes de santé publique et individuelle.

Après avoir exploré mon parcours au sein du CHU de Bejaia et les défis rencontrés dans le premier chapitre, nous plongeons maintenant dans une analyse approfondie de la science des données appliquée au domaine de la santé. L'intersection entre ces deux domaines est d'une importance cruciale, car elle représente l'avenir de la médecine personnalisée et de la prise de décision basée sur les données. Dans ce contexte, nous allons examiner de près la démarche d'un data scientist, depuis la collecte et la préparation des données jusqu'à la construction et l'évaluation de modèles sophistiqués.

2.1 La science des données et la santé

Selon une étude, chaque corps humain génère 2 téraoctets de données par jour. Ces informations incluent l'activité cérébrale, le stress, le sucre, la fréquence cardiaque et bien d'autres choses. Pour gérer et maintenir de telles quantités de données, nous disposons désormais de technologies plus avancées, dont l'une est la science des données. Elle aide à suivre la santé des patients en utilisant des données enregistrées.

Grâce aux applications de la science des données dans le domaine de la santé, il est désormais possible de détecter les symptômes de la maladie à un stade précoce. Les médecins peuvent également surveiller l'état des patients à distance grâce au développement de divers outils et technologies révolutionnaires.

Autrefois, les médecins et l'administration hospitalière étaient incapables de prendre en charge un grand nombre de patients simultanément. En raison d'un manque de traitement adéquat, l'état des patients s'est dégradé.

Avec les applications de la science des données dans le domaine de la santé, la situation a changé. Les applications de science des données et d'apprentissage automatique peuvent informer les médecins de l'état de santé des patients via des appareils portables. L'administration hospitalière peut alors dépêcher des médecins stagiaires, des assistants ou du personnel infirmier au domicile de ces patients.

Les hôpitaux peuvent également installer divers équipements et appareils de diagnostic pour ces patients. Ces appareils basés sur la science peuvent collecter des données sur les patients telles que la fréquence cardiaque, la pression artérielle, la température, etc. Les mises à niveau et les notifications dans les applications mobiles fournissent aux médecins des données sur la santé des patients en temps réel. Ils peuvent ensuite diagnostiquer les conditions et aider les jeunes médecins ou infirmières à administrer des traitements spécifiques aux patients à leur

domicile. C'est ainsi que la science des données peut faciliter les soins aux patients en utilisant la technologie.[20]

2.2 Avantages de la science des données dans le secteur de la santé

Pour les soins de santé, la science des données est désormais un élément crucial qui a révolutionné l'industrie. De nombreuses avancées ont été possibles grâce à ses outils technologiques et à ses techniques, accélérant ainsi les traitements et les diagnostics. En conséquence, le flux de travail dans le domaine de la santé s'est nettement amélioré.
Voici quelque avantages de la science des données dans le domaine de la santé :

- Elle améliore l'efficacité du flux de travail des soins de santé.
- Elle offre un traitement rapide et approprié.
- Elle aide à la bonne gestion des situations d'urgence.
- Elle raccourcit le temps de traitement pour les patients.
- Elle aide à réduire le risque d'échec lors du traitement de toute personne concernée.

On peut légitimement se questionner sur l'intégration de la data science dans le domaine de la santé et sur la manière dont elle est appliquée. Parmi les nombreux domaines d'application, on retrouve l'imagerie médicale, l'analyse des données génomiques, la recherche de nouveaux médicaments, l'analyse prédictive de la santé, la surveillance des patients, ainsi que la prévention et la surveillance des maladies et ainsi de suite.

2.3 Tendances futures de la science des données dans le secteur de la santé

Maintenant que nous avons fait un tour sur la science des données et le domaine de la santé, jetons maintenant un coup d'œil aux quatre facteurs qui entraînent une amélioration spectaculaire de l'industrie de la santé.

- Besoin de digitalisation.
- Innovations technologiques.
- La nécessité de traiter avec une population nombreuse.
- Coûts de traitement élevés.

L'avenir de la science des données dans le domaine de la santé sera guidé par l'essor de l'intelligence artificielle (IA) et de la technologie d'apprentissage automatique (ML). Ces deux technologies révolutionnent déjà de nombreux secteurs, de la finance au commerce de détail, il n'est donc pas surprenant qu'elles trouvent également leur place dans le secteur de la santé.

2.4 Démarche d'un data scientist

La Data Science est en plein essor, avec une demande croissante de candidats et de formations dans ce domaine. L'émergence de Data Labs dans les organisations montre l'importance de cette science pour la prise de décision et l'optimisation des processus métiers. Le Big Data, les technologies comme Hadoop ou Spark, et les avancées du machine learning expliquent cet engouement. Un projet Data Science comprend plusieurs étapes : définition des objectifs, collecte et nettoyage des données, construction d'hypothèses, modélisation prédictive et présentation des résultats [5].

Voici les 7 étapes d'un projet d'un data science illustrées dans la figure 2.1.

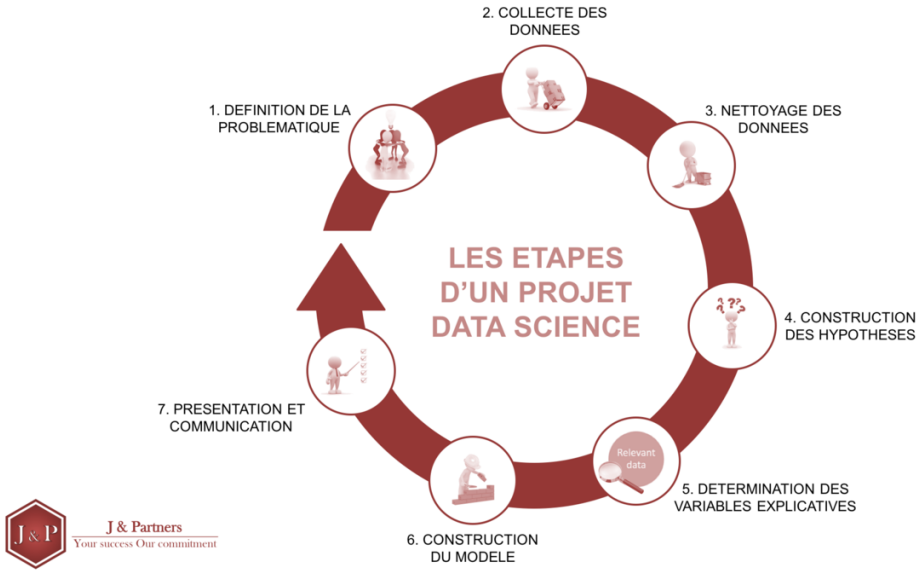


Figure 2.1: Démarche d'un projet en science des données

2.4.1 Compréhension du problème métier

Le data scientist vise à utiliser des données de qualité pour orienter les décisions d'une entreprise. Avant de débuter un projet, il doit comprendre l'environnement et définir la problématique. Il répond à cinq types de questions : quantité (régression), choix entre A et B (classification), organisation des données (clustering), détection d'anomalies, et prochaines actions à prendre. La collaboration avec les responsables métiers est cruciale pour comprendre la problématique et identifier les variables à prédire.

La compréhension des données et la capacité à poser les bonnes questions sont des aspects essentiels, souvent acquis par l'expérience et l'échange avec des experts [5].

2.4.2 Collecte des données

Une fois les objectifs du projet bien définis, il est alors temps de collecter les données.

La collecte de données pour un data scientist est le processus de rassemblement et d'acquisition de différentes sources de données pertinentes pour un projet donné.

Malheureusement La collecte de données est souvent complexe, loin de l'idée naïve d'avoir toutes les données facilement disponibles. Cela nécessite du temps et de l'énergie, le data scientist doit avoir une vision claire et exhaustive des données à collecter, identifier les sources où obtenir ces données, savoir y accéder et les stocker [5].

2.4.3 Nettoyage des données

Après avoir la collecte des données le data scientist passe à l'étape la plus chronophage du projet (50 à 80% du temps),

L'étape de préparation des données est cruciale non seulement en raison du volume de données, mais aussi du temps qu'elle nécessite, principalement en raison des interactions fréquentes entre le data scientist et les métiers impliqués. Une compréhension approfondie de l'environnement étudié est essentielle pour éviter les omissions d'informations et ainsi prévenir les biais dans l'analyse des données. Avec l'avènement des projets Big Data, la quantité de données à traiter devient de plus en plus conséquente, atteignant parfois plusieurs téraoctets de données.

Il est donc crucial de comprendre les aspects de cette étape. Les données proviennent de diverses sources et peuvent présenter différents formats (*csv, json, xml, etc.*) ainsi que des anomalies ou des valeurs incorrectes. Les problèmes de qualité les plus courants incluent:

- Les données erronées : généralement dues à des erreurs de saisie ou des incompatibilités entre la source et la base de données.
- Les données incomplètes : fréquemment rencontrées lorsque seuls les champs obligatoires ou ceux pertinents pour les utilisateurs sont renseignés.
- Les données non normalisées : plusieurs formats différents sont utilisés pour représenter des données identiques, comme les différents codes pour indiquer le sexe d'une personne.
- Les données obsolètes : des informations périmées qui peuvent affecter la qualité de la base de données.
- Les doublons : des entrées répétées dans la base de données, nécessitant une fusion pour éviter les erreurs dans l'analyse [5].

Ces problèmes peuvent entraîner des erreurs dans la formulation des hypothèses et des biais lors de la construction des modèles prédictifs.

Pour nettoyer les données, le data scientist utilise divers outils et techniques. Par exemple, il peut utiliser KNIME, Python (avec la bibliothèque Pandas) ou R (avec Data.table). Ils suppriment ou remplacent les données manquantes, effectuent des opérations de filtrage, tri, regroupement, fusion et pivotement. L'objectif est de préparer les données pour faciliter l'exploration, la formulation des hypothèses et l'entraînement des modèles de machine learning [5].

2.4.4 Formulation des hypothèses

Le data scientist, une fois en possession d'un jeu de données complet et bien préparé, commence l'analyse en effectuant un exercice de brainstorming. L'objectif est de découvrir des corrélations

entre les différentes données, ce qui se traduit par la formulation d'hypothèses. Par exemple, pour estimer le prix de vente d'un bien immobilier, les données pertinentes incluent la localisation, la superficie, le rendement locatif, l'âge et la qualité de construction, les équipements, etc. Cette analyse est soutenue par l'utilisation d'histogrammes, de courbes de distribution et de diagrammes de dispersion pour identifier les tendances. Des outils comme Power BI ou QlikView facilitent ce processus avec des visualisations interactives. Il est important de noter que cette phase descriptive est itérative et se déroule en parallèle avec le nettoyage des données, qui révèle les incohérences [5].

2.4.5 Détermination des variables synthétiques

Le data scientist, après avoir formulé des hypothèses et identifié les variables impactant la variable cible, passe à l'étape du "feature engineering". Cela implique la création et la sélection de variables synthétiques qui améliorent la représentation du problème à résoudre et la performance du modèle.

Le processus inclut la sélection des variables pertinentes pour simplifier les modèles et réduire la dimensionnalité, ainsi que la transformation des variables pour créer de nouvelles variables basées sur des seuils ou des critères spécifiques. Par exemple, si une variable brute représente l'âge et que le modèle de prédiction serait plus performant en se basant sur un indicateur de majorité, le data scientist pourrait définir un seuil à 18 ans. Cela lui permettrait de créer deux nouvelles variables, "majeur" et "mineur", en fonction de cette distinction [5].

2.4.6 Construction du modèle

L'étape de construction de modèle est une phase centrale dans le processus de data science et d'apprentissage automatique.

Il s'agit de choisir les différents modèles de machine learning qui permettent de modéliser au mieux la variable cible à expliquer (problématique métier) [5].

2.4.7 Présentation et Communication

Une fois le modèle prédictif établi et validé, il est temps de communiquer sur ses résultats. le data scientist devra, avec l'aide des équipes IT, industrialiser sa solution et l'intégrer dans l'infrastructure existante de l'entreprise [5].

2.5 Vue d'ensemble du Machine Learning

Lorsque le terme "Machine Learning" (ML), traduit en français par "apprentissage automatique", est évoqué, beaucoup de gens imaginent souvent un robot: un serviteur fiable ou un Terminator redoutable, selon l'interlocuteur. Cependant, le Machine Learning n'est pas un concept futuriste ; c'est déjà une réalité. En fait, il est présent depuis des décennies dans certaines applications spécialisées telles que la reconnaissance optique de caractères ou OCR. La première application ML ayant véritablement touché un large public, améliorant le quotidien de centaines de millions de personnes, s'est imposée dans les années 1990 .La question est de savoir où se

situe la limite de l'apprentissage automatique. Quand on dit qu'une machine apprend, que cela implique-t-il? [2]

2.5.1 Qu'est-ce que l'apprentissage automatique ?

L'apprentissage automatique est un domaine qui combine la science et l'art de développer des modèles et des algorithmes permettant aux ordinateurs d'apprendre à partir de données.

Voici un définition plus générale :

L'apprentissage automatique est la] discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés. - Arthur Samuel, 1959-

En voici une autre plus technique :

Étant donné une tâche T et une mesure de performance P, on dit qu'un programme informatique apprend à partir d'une expérience E si les résultats obtenus sur T, mesurés par P, s'améliorent avec l'expérience E. -Tom Mitchell, 1997- [10]

2.5.2 Pourquoi utiliser l'apprentissage automatique ?

L'apprentissage automatique est une discipline qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés. Cela implique le développement de modèles et d'algorithmes qui identifient des schémas et des structures dans les données, leur permettant de faire des prédictions ou des recommandations. Cette approche trouve des applications dans divers domaines comme la santé, la finance et la sécurité, où elle est utilisée pour la détection de maladies, l'analyse des risques et la détection de fraudes, entre autres.
Pourquoi utilise-t-on le Machine Learning ?

L'apprentissage automatique offre une palette de solutions pour résoudre des problèmes divers:

- Il intervient lorsque nous sommes confrontés à des problèmes pour lesquels nous n'avons pas de solution directe, comme la prédiction d'achats futurs.
- Il est également utile pour résoudre des problèmes que nous savons déjà résoudre, mais pour lesquels nous ne pouvons pas formaliser les solutions en termes algorithmiques, comme la reconnaissance d'images ou la compréhension du langage naturel.
- De plus, il nous aide à résoudre des problèmes pour lesquels les procédures traditionnelles sont trop gourmandes en ressources informatiques, comme la prédiction d'interactions entre de grandes molécules.

Dans ces cas, ML est particulièrement efficace lorsque les données sont abondantes mais que nos connaissances pour les traiter sont limitées ou peu développées.

Par ailleurs, ML peut également être un atout précieux pour l'apprentissage humain. Les modèles créés par ces algorithmes peuvent révéler l'importance relative des différentes informations et la manière dont elles interagissent pour résoudre des problèmes spécifiques. Par exemple, dans le domaine de la prédiction d'achats, comprendre le modèle peut nous aider à analyser les caractéristiques des achats passés qui influencent les futurs. Cet aspect du Machine Learning est largement utilisé dans la recherche scientifique pour comprendre des phénomènes complexes comme les interactions génétiques dans le développement de certaines tumeurs, les régions cérébrales liées à certains comportements, les caractéristiques moléculaires des médicaments efficaces, ou encore l'identification d'objets astronomiques à partir d'images de télescopes [10].

2.5.3 Types de systèmes d'apprentissage automatique

L'apprentissage automatique est un champ assez vaste, et nous dressons dans cette section une liste des plus grandes classes de problèmes auxquels il s'intéresse, voir la figure 2.2:

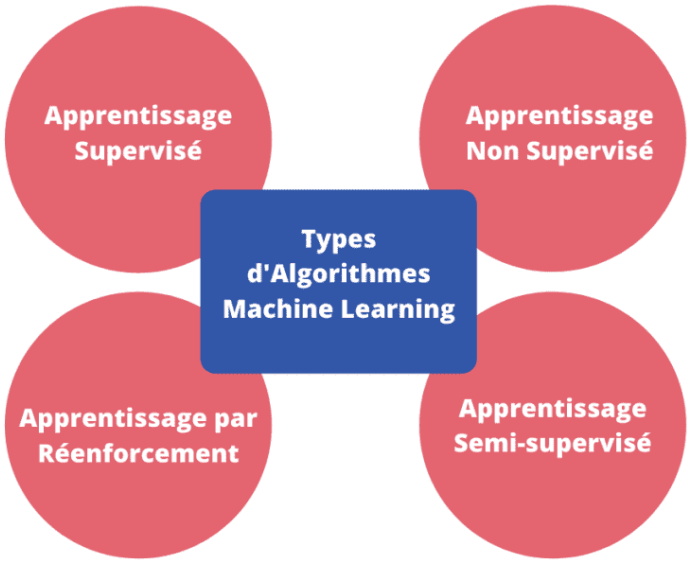


Figure 2.2: Types d'apprentissage automatique

Apprentissage supervisé

L'apprentissage automatique supervisé est un modèle qui s'appuie dans la phase de formation sur des données classifiées et des exemples clairement étiquetés, contenant des données d'entrée et de sortie qui sont utilisées pour former la machine. L'objectif principal de la phase de formation est de permettre à la machine de comprendre la relation entre les données d'entrée et de sortie. Si la machine apprend la relation entre les données d'entrée et de sortie, nous pouvons l'utiliser pour classer les données nouvelles et différentes. Les utilisations les plus importantes de l'apprentissage supervisé sont l'évaluation des risques, la classification des images, la détection des fraudes, le filtrage des spams, etc dematteis.

L'apprentissage supervisé consiste à établir des règles de comportement à partir d'une base de données contenant des exemples de cas déjà étiquetés. Plus précisément, cette base de

données est un ensemble de couples entrées-sorties (Xi, Yi) sublime2022 choisis au hasard. L'objectif est alors d'apprendre à prédire, pour toute nouvelle entrée X, la sortie Y. La figure suivante montre un exemple d'apprentissage supervisé.

Un des exemples les plus courants de l'apprentissage supervisé est la reconnaissance des types d'animaux, où nous apprenons à la machine comment traiter des milliers d'images pour différents types d'animaux. Lors de la phase de test, nous introduisons de nouvelles images et cela nous donne le type d'image. Voir la figure 2.3.

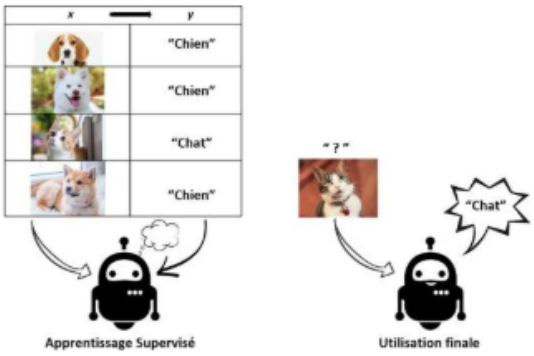


Figure 2.3: Exemple d'apprentissage supervisée

Apprentissage non supervisé

En apprentissage supervisé, nous cherchons à entraîner un modèle capable de mapper une entrée à une sortie après avoir appris certaines caractéristiques, acquérant ainsi une capacité de généralisation pour classer correctement des échantillons de données jamais vus. Cependant, il arrive parfois que nous ne connaissions pas la sortie, car nous n'avons que les données d'entrée et ne pouvons pas définir une étiquette de sortie pour chaque échantillon d'entrée.

Supposons que nous travaillons pour une entreprise de vente de vêtements et que nous disposons de données sur les clients précédents: combien ils ont dépensé, leur âge et le jour où ils ont acheté le produit. Notre tâche consiste à trouver un modèle ou une relation entre les variables afin de fournir à l'entreprise des informations utiles pour qu'elle puisse créer des stratégies marketing, décider sur quel type de client elle devrait se concentrer pour maximiser les profits ou sur quel segment de clients elle peut investir davantage pour se développer sur le marché [11].

Voici quelques-uns des algorithmes d'apprentissage non supervisé les plus importants :

- Clustering :
  - K-Means
  - Analyse des clusters hiérarchiques (HCA)
  - Maximisation des attentes
- Visualisation et réduction de la dimensionalité :

  - Analyse en composantes principales (ACP)
  - Kernel PCA
  - L'encastrement linéaire local (LLE)
  - T-distribué Stochastic Neighbor Embedding (t-SNE)

- Apprentissage des règles d'association :
  - Apriori
  - Eclat

Apprentissage semi-supervisé

Comme on peut s'en douter, l'apprentissage semi-supervisé consiste à apprendre des étiquettes à partir d'un jeu de données partiellement étiqueté. Le premier avantage de cette approche est qu'elle permet d'éviter d'avoir à étiqueter l'intégralité des exemples d'apprentissage, ce qui est pertinent quand il est facile d'accumuler des données mais que leur étiquetage requiert une certaine quantité de travail humain.

Prenons par exemple la classification d'images, il est facile d'obtenir une banque de données contenant des centaines de milliers d'images, mais avoir pour chacune d'entre elles l'étiquette qui nous intéresse peut requérir énormément de travail. De plus, les étiquettes données par des humains sont susceptibles de reproduire des biais humains, qu'un algorithme entièrement supervisé reproduira à son tour. L'apprentissage semi-supervisé permet parfois d'éviter cet écueil. Il s'agit d'un sujet plus avancé, que nous ne considérerons pas dans cet ouvrage [2].

Apprentissage par renforcement

Dans le cadre de l'apprentissage par renforcement, le système d'apprentissage peut interagir avec son environnement et accomplir des actions. En retour de ces actions, il obtient une récompense, qui peut être positive si l'action était un bon choix, ou négative dans le cas contraire. La récompense peut parfois venir après une longue suite d'actions; c'est le cas par exemple pour un système apprenant à jouer au go ou aux échecs. Ainsi, l'apprentissage consiste dans ce cas à définir une politique, c'est-à-dire une stratégie permettant d'obtenir systématiquement la meilleure récompense possible.

Les applications principales de l'apprentissage par renforcement se trouvent dans les jeux (échecs, go, etc) et la robotique. Ce sujet dépasse largement le cadre de cet ouvrage [2].

2.5.4 L'apprentissage automatique au service de la santé

L'apprentissage automatique révolutionne le domaine de la santé en offrant des solutions novatrices pour le diagnostic, l'accompagnement des patients et l'optimisation des processus hospitaliers. Il permet aux professionnels de santé de gagner du temps et d'améliorer la précision des diagnostics en analysant de vastes quantités de données médicales. Les applications vont de la détection précoce des maladies à l'assistance chirurgicale de pointe.

Par exemple, des algorithmes d'apprentissage automatique peuvent détecter des anomalies sur des images médicales 1000 fois plus rapidement que les humains, contribuant ainsi à des soins plus rapides et précis.

Les assistants virtuels basés sur le machine learning, comme les chatbots ou applications dédiées, se déploient également pour accompagner les patients tout au long de leur parcours de soins, répondant instantanément à leurs questions et contribuant à l'observance thérapeutique. Parallèlement, en chirurgie, l'intelligence artificielle assiste les chirurgiens pour améliorer la précision des gestes et réduire les erreurs, notamment à travers des robots chirurgiens avancés.

L'automatisation des tâches administratives via le machine learning simplifie également la gestion hospitalière, permettant aux professionnels de santé de se concentrer sur les soins et d'optimiser les coûts associés au parcours du patient à l'hôpital [18].

2.5.5 Principales difficultés de l'apprentissage automatique

Qualité des données

L'un des problèmes les plus fondamentaux de l'apprentissage automatique est d'assurer la qualité des données utilisées pour entraîner et tester les modèles d'apprentissage automatique. Les praticiens doivent donc nettoyer, valider et augmenter les données pour garantir leur exactitude, leur exhaustivité et leur pertinence aux problèmes spécifiques. Cela garantit que les modèles reflètent fidèlement les scénarios réels et produisent des résultats fiables pour les utilisateurs et les entreprises [10].

Sélection de problème

Le choix du modèle en apprentissage automatique est crucial car chaque type de modèle a ses forces, faiblesses et hypothèses spécifiques. Utiliser le mauvais modèle peut conduire à des performances médiocres, du sur-ajustement ou du sous-ajustement. Ainsi, les praticiens doivent comprendre le problème, les données et les caractéristiques des modèles pour choisir judicieusement [10].

Interprétabilité du modèle

La compréhension de la manière dont les modèles fonctionnent et la justification de leurs prédictions ou décisions sont des défis importants en apprentissage automatique, surtout pour les modèles complexes comme les réseaux neuronaux profonds. L'interprétabilité des modèles est cruciale pour garantir la confiance, la transparence et la responsabilité, en particulier dans des domaines critiques comme la santé ou la finance. Les praticiens doivent donc recourir à des techniques telles que l'importance des caractéristiques ou les diagrammes de dépendance partielle pour expliquer le comportement et la logique des modèles [10].

Généralisation du modèle

Un autre problème de l'apprentissage automatique est de s'assurer que les modèles peuvent bien fonctionner avec de nouvelles données qui peuvent être différentes de celles sur lesquelles ils

ont été entraînés ou testés. Cela peut inclure des distributions différentes, des caractéristiques variées ou même des cas rares et nouveaux. La généralisation des modèles est cruciale pour assurer leur robustesse et fiabilité, surtout dans des environnements changeants. Les experts en apprentissage automatique utilisent des techniques comme la validation croisée, la régularisation et l'augmentation des données pour éviter les problèmes de surajustement ou de sous-ajustement et pour rendre les modèles plus adaptables [10].

Déploiement du modèle

Le dernier problème du ML est le déploiement des modèles dans des systèmes de production qui peuvent répondre aux besoins des utilisateurs ou de l'entreprise. Les modèles doivent répondre à diverses exigences telles que la vitesse, l'évolutivité, la sécurité et la compatibilité, ce qui peut dépasser les capacités des infrastructures existantes. De plus, ils nécessitent une surveillance, des mises à jour et parfois un réentraînement pour rester pertinents.

Le déploiement efficace des modèles est crucial pour tirer parti de leur valeur dans les applications réelles. Les professionnels du ML utilisent des outils tels que Docker, Kubernetes ou AWS pour faciliter ces processus de déploiement et de gestion des modèles [10].

2.5.6 Algorithmes d'apprentissage automatique

Les algorithmes d'apprentissage automatique permettent aux systèmes informatiques de reconnaître des motifs et de prendre des décisions basées sur des données. Ces techniques sont cruciales dans divers domaines, allant de la reconnaissance d'images à la prévision financière. Les deux principales catégories d'algorithmes d'apprentissage supervisé sont la classification et la régression, chacune ayant des applications spécifiques et des méthodes distinctes pour résoudre des problèmes complexes.

Classification

La classification est une technique d'apprentissage automatique qui consiste à entraîner un modèle à attribuer une étiquette de classe à une entrée donnée. Il s'agit d'une tâche d'apprentissage supervisé, ce qui signifie que le modèle est formé sur un ensemble de données étiquetées qui comprend des exemples de données d'entrée et les étiquettes de classe correspondantes.

Le modèle vise à apprendre la relation entre les données d'entrée et les étiquettes de classe afin de prédire l'étiquette de classe pour de nouvelles entrées inédites.

Régression

La régression est un type d'apprentissage supervisé dont l'objectif est de prédire une variable dépendante basée sur une ou plusieurs caractéristiques d'entrée (également appelées prédicteurs ou variables indépendantes).

Les algorithmes de régression sont utilisés pour modéliser la relation entre les entrées et les sorties et faire des prédictions basées sur cette relation. La régression peut être utilisée pour des variables dépendantes continues ou catégorielles.

En général, l'objectif de la régression est de construire un modèle qui peut prédire avec précision la sortie sur la base des caractéristiques d'entrée et de comprendre la relation sous-jacente entre les caractéristiques d'entrée et la sortie.

Les forêts aléatoires

Les forêts aléatoires sont un algorithme de ML conçu pour obtenir une prédiction fiable grâce à un système de sous-espaces aléatoires. Elles sont composées de plusieurs arbres de décision, entraînés de manière indépendante sur des sous-ensembles du dataset d'apprentissage (méthode de bagging). Chacun produit une estimation; la moyenne (ou le vote, dans le cas d'un problème de classification) de tous les arbres est la prédiction finale. Elles sont utilisées pour réduire la variance des prévisions d'un arbre de décision seul, améliorant ainsi leurs performances. Les forêts aléatoires ont été proposées par Leo Breiman et Adele Cutler en 2001 [12].

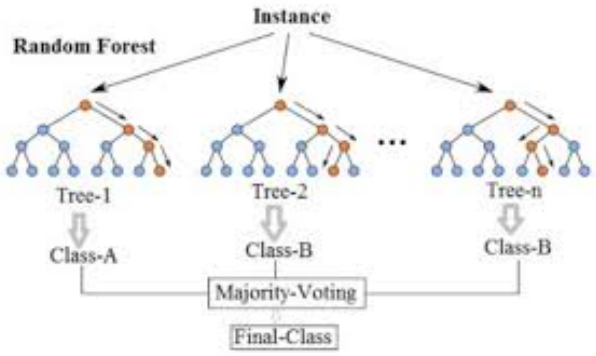


Figure 2.4: Foret aléatoire.

Ridge Regression

La régression Ridge, également appelée régression de crête, est une méthode d'estimation utilisée en régression linéaire pour traiter les problèmes de colinéarité (où les variables prédictives sont très fortement corrélées entre elles). Elle ajoute un terme de pénalité de régularisation L2 à la fonction de coût de la régression linéaire standard. Cette régularisation aide à réduire la variance des estimations et à améliorer la stabilité et la prédictibilité du modèle.

La formule de la régression Ridge est la suivante :

$$\text{minimize} \left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right)$$

où :

- y_i sont les valeurs observées,
- $\hat{y}_i$ sont les valeurs prédites,
- β_j sont les coefficients de régression,

- α est le paramètre de régularisation.

Le terme de régularisation L2 ($\alpha \sum_{j=1}^{p} \beta_j^2$) pénalise la somme des carrés des coefficients de régression, ce qui empêche les coefficients de devenir trop grands et aide à contrôler la complexité du modèle [16].

Conclusion

Dans ce chapitre, nous avons définit la science des données et sa relation avec la santé, ainsi que les étapes nécessaires pour résoudre un problème à l'aide du ML de manière générale.

Dans le prochain chapitre, nous explorerons les données existantes au sein du CHU en vue de créer un modèle permettant d'atteindre l'objectif fixé.

3

Déroulement du cas pratique

Introduction

Dans le chapitre précédent, des concepts essentiels ont été définis pour acquérir les connaissances nécessaires à la compréhension du sujet. Dans ce chapitre, la résolution du problème sera abordée en utilisant des données. Chaque étape de ce processus sera interprétée, avec une définition de logiciel et des packages utilisés.

3.1 Python

3.1.1 Définition

Python est un langage de programmation interprété, orienté objet et polyvalent, avec une syntaxe claire et une grande puissance. Il supporte la programmation procédurale et fonctionnelle, ainsi que le typage dynamique. Python est portable sur diverses plateformes comme Unix, Linux, macOS et Windows, et peut être étendu en C/C++. Python offre une gamme étendue de bibliothèques puissantes qui enrichissent son écosystème et en font un choix privilégié pour les data scientists [4].

Voici quelques bibliothèques qu'il faut connaître pour un data scientist:

Le package Numpy: Le module NumPy propose également des fonctions essentielles pour effectuer des opérations mathématiques avancées, telles que l'algèbre linéaire, qui est utilisée pour résoudre des systèmes d'équations linéaires et d'autres problèmes mathématiques complexes [4].

Le package pandas: Pandas est une bibliothèque Python utilisée pour manipuler, analyser et nettoyer des ensembles de données. Créée par Wes McKinney en 2008, son nom vient de "Panel Data" et "Python Data Analysis". Elle offre des fonctions pour explorer les données, trouver des corrélations, calculer des statistiques comme la moyenne, la valeur maximale et minimale. Pandas permet également le nettoyage des données en supprimant les lignes non pertinentes ou contenant des valeurs erronées [4].

Le package Matplotlib: Matplotlib est une bibliothèque Python open source créée par John D. Hunter pour le traçage et la visualisation de graphiques. Elle est principalement écrite en Python, avec quelques parties en C, Objective-C et JavaScript pour assurer la compatibilité multiplateforme.

Il est également nécessaire de maîtriser d'autres bibliothèques en Python telles que Seaborn, Tkinter, Scipy, et d'autres encore. De plus, il est important d'utiliser les fonctions nécessaires à chaque modèle pour garantir des résultats précis et efficaces [4].

3.2 Déroulement du cas pratique

La démarche d'un data scientist, définie dans le deuxième chapitre, constitue le cadre pour les étapes suivies dans le cas pratique.

3.2.1 Compréhension du problème métier

L'étape de compréhension des données est la première phase d'un projet. Dans mon cas, il s'agit de mieux connaître, de manière générale, les données de l'hôpital et de prédire un cas dans ce système de gestion des patients.

3.2.2 Collectes des données

La collecte des données, généralement l'étape la plus longue, a été facilitée dans ce projet grâce à l'utilisation directe du logiciel Patient. Cette méthode a permis d'obtenir toutes les données disponibles au niveau du bureau des entrées.

3.2.3 Nettoyage des données

Étape de nettoyage des données : Début de la manipulation des données avec Python. Importation des bibliothèques nécessaires, notamment NumPy et pandas. Affichage de la base de données nommée F30.

Voici la figure 3.1 qui représente la base de données récupérée.

	FER_ENTREE	FER_MOTP	FER_DAT_EN	FER_HEURE	FER_COND	FER_SERV	FER_SALLE	FER_LIT	FER_NOM_MA	FER_PRE_MA	...	FER_DT_SOI
0	2301000001	PATIEN	2023-01-01	12.0	4	232.0	3.0	0	AMARA	TAOUS	...	2023-01-1
1	2301000002	PATIEN	2023-01-01	16.0	1	232.0	3.0	0	AIT MOUSSA	MELISSA	...	2023-01-0
2	2301000003	PATIEN	2023-01-01	51.0	4	232.0	3.0	0	BEN ZIANE	RADIA	...	2023-02-0
3	2301000004	PATIEN	2023-01-01	108.0	4	232.0	3.0	0	HAMANI	AMEL	...	2023-01-0
4	2301000005	PATIEN	2023-01-01	15.0	5	232.0	3.0	0	OUARI	ABDERAHMANE	...	2023-01-0
...	...	...	...	...	...	...	...	...	...	...	...	.
26189	2301017339	PATIEN	2023-12-31	2246.0	4	232.0	3.0	0	MESSAOUDI	HANANE	...	Na
26190	2301017340	PATIEN	2023-12-31	2314.0	4	232.0	3.0	0	GUECHTAL	HASSIBA	...	Na
26191	2301048906	PATIEN	2023-12-31	2320.0	1	112.0	2.0	0	AKLI	A/RAHMANE	...	Na
26192	2301048907	PATIEN	2023-12-31	2335.0	1	221.0	4.0	0	BENSADALLAH	MOHAMED LAMINE	...	Na
26193	2301017341	PATIEN	2023-12-31	2110.0	5	232.0	3.0	0	AMIRA	ZINEB INES	...	Na

26194 rows × 54 columns

Figure 3.1: Dataframe

Le DataFrame contient 26 193 lignes et 54 colonnes.

Importer *matplotlib* et *seaborn* est important pour visualiser les données et dessiner des graphiques afin de mieux comprendre les données.

Analyse de la forme

Avant de commencer le nettoyage, il est important de bien connaître mes données et de les analyser.

```
# Analyse de la forme des données
data=df.copy()
```

Figure 3.2: Analyse de la forme

Ce code nous permet de créer une copie de notre DataFrame original, et cela permet de travailler avec **df** et de garder une copie **data**.

```
: df.columns

: Index(['FER_ENTREE', 'FER_MOTP', 'FER_DAT_EN', 'FER_HEURE', 'FER_COND',
         'FER_SERV', 'FER_SALLE', 'FER_LIT', 'FER_NOM_MA', 'FER_PRE_MA',
         'FER_SEXE', 'FER_TYP_DT', 'FER_DAT_NA', 'FER_LIEU_N', 'FER_FILS',
         'FER_ET_N', 'FER_ET_P', 'FER_ADR_MA', 'FER_POS_MA', 'FER_SIT_FA',
         'FER_EP_N', 'FER_EP_P', 'FER_NATION', 'FER_TELM', 'FER_CSP', 'FER_PROF',
         'FER_AS_DEM', 'FER_PA_CN', 'FER_PA_CP', 'FER_ADR_C', 'FER_POS_C',
         'FER_ACCOMP', 'FER_ACOMP2', 'FER_NOM_ME', 'FER_MED_TR', 'FER_ETAB',
         'FER_G_SANG', 'FER_ANCNUM', 'M_TYP_PID', 'M_NUM_PID', 'M_DT_DLV',
         'M_LIEU_D1', 'M_LIEU_D2', 'DT_SOR_MED', 'FER_DT_SOR', 'HEURE_SORT',
         'FER_MOD_SO', 'FER_TYP_DC', 'NUM_DEC', 'HEURE_DEC', 'FER_DIAG_E',
         'FER_DIAG_S', 'NOM_MED_SO', 'FER_MOTP_S'],
        dtype='object')
```

Figure 3.3: Toutes les colonnes du Dataframe

La fonction **df.columns** permet d'afficher toutes les colonnes.
Voici l'explication de chaque colonne :

- **FER_ENTREE (int64)** : Un identifiant unique pour chaque entrée d'hôpital.
- **FER_MOTP (object)** : Motif principal d'admission (probablement un code ou une description textuelle).
- **FER_DAT_EN (object)** : Date d'entrée à l'hôpital (stockée sous forme de chaîne de caractères).
- **FER_HEURE (float64)** : Heure d'entrée à l'hôpital.
- **FER_COND (int64)** : Code conditionnel (probablement lié à l'état du patient à l'entrée).
- **FER_SERV (float64)** : Code du service où le patient a été admis.
- **FER_SALLE (float64)** : Numéro de la salle où le patient a été admis.
- **FER_LIT (int64)** : Numéro du lit attribué au patient.
- **FER_NOM_MA (object)** : Nom de famille du patient.
- **FER_PRE_MA (object)** : Prénom du patient.
- **FER_SEXE (object)** : Sexe du patient.
- **FER_TYP_DT (object)** : Type de document (probablement lié à l'admission).
- **FER_DAT_NA (object)** : Date de naissance du patient.
- **FER_LIEU_N (object)** : Lieu de naissance du patient.
- **FER_FILS (object)** : Filiation (informations sur les parents).

- **FER_ET_N (object)** : État civil du patient.
- **FER_ADR_MA (object)** : Adresse principale du patient.
- **FER_POS_MA (object)** : Code postal de l'adresse principale.
- **FER_SIT_FA (object)** : Situation familiale du patient.
- **FER_ET_P (object)** : État du patient.
- **FER_PROF (object)** : Profession du patient.
- **FER_AS_DEM (object)** : Assurance demandée.
- **FER_EP_N (object)** : Nom de l'époux/épouse.
- **FER_EP_P (object)** : Prénom de l'époux/épouse.
- **FER_NATION (object)** : Nationalité du patient.
- **FER_TEL_M (object)** : Numéro de téléphone mobile.
- **FER_CSP (float64)** : Catégorie socioprofessionnelle (probablement un code).
- **FER_PA_CN (object)** : Code postal de l'adresse de contact.
- **FER_PA_CP (object)** : Code postal de l'adresse professionnelle.
- **FER_ADR_C (object)** : Adresse de contact secondaire.
- **FER_POS_C (object)** : Code postal de l'adresse de contact secondaire.
- **FER_ACCOMP (object)** : Nom de l'accompagnateur.
- **FER_ACOMP2 (object)** : Autre accompagnateur.
- **FER_NOM_ME (float64)** : Code du médecin traitant.
- **FER_MED_TR (object)** : Nom du médecin traitant.
- **FER_ETAB (float64)** : Code de l'établissement de santé.
- **FER_G_SANG (object)** : Groupe sanguin du patient.
- **FER_ANCNUM (float64)** : Ancien numéro de dossier.
- **M_TYP_PID (float64)** : Type d'identifiant patient.
- **M_NUM_PID (object)** : Numéro d'identifiant patient.
- **M_DT_DLV (object)** : Date de délivrance du document d'identité.
- **M_LIEU_D1 (float64)** : Lieu de délivrance (code).
- **M_LIEU_D2 (float64)** : Lieu de délivrance secondaire (inutilisé).

- **DT_SOR_MED (object)** : Date de sortie médicale.
- **FER_DT_SOR (object)** : Date de sortie administrative.
- **HEURE_SORT (object)** : Heure de sortie.
- **FER_MOD_SO (float64)** : Mode de sortie (code).
- **FER_TYP_DC (object)** : Type de décharge (document de sortie).
- **NUM_DEC (object)** : Numéro de déclaration.
- **HEURE_DEC (float64)** : Heure de déclaration.
- **FER_DIAG_E (object)** : Diagnostic à l'entrée.
- **FER_DIAG_S (object)** : Diagnostic à la sortie (inutilisé).
- **NOM_MED_SO (object)** : Nom du médecin de sortie (inutilisé).
- **FER_MOTP_S (object)** : Motif de sortie.

```
dtypes: datetime64[ns](5), float64(14), int64(3), object(32)
memory usage: 10.8+ MB
```

Figure 3.4: Les infos sur types des données

Cette capture explique très bien les types de données, ainsi que l'utilisation de la mémoire.
Remarque : L'importance de la vérification des données réside dans la possibilité de travailler avec des données incorrectement implémentées. Cette situation impose une vérification minutieuse des données. Un échantillon est prélevé pour une vérification du type de données, les erreurs étant souvent liées à la saisie des dates.

```
df['FER_DT_SOR'].unique()
```

Figure 3.5: Code affiche la colonne FER_DT_DOR

```
 '2023-12-26T00:00:00.000000000', '2023-12-16T00:00:00.000000000',
 '2023-12-25T00:00:00.000000000', '2023-12-23T00:00:00.000000000',
 '2023-12-29T00:00:00.000000000', '2023-12-31T00:00:00.000000000'],
dtype='datetime64[ns]')
```

Figure 3.6: FER_HEURE

Vérification de la colonne `'FER_HEURE'` : Utilisation du code df[`''FER_HEURE''`].unique() pour identifier le problème de type de données, qui est une chaîne de caractères. Conversion nécessaire en type temporel (datetime) à l'aide de techniques et de fonctions spécifiques. L'image suivante montre le code utilisé pour convertir certaines colonnes en datetime.

```
from datetime import datetime
dh = pd.DataFrame(df)

# Fonction pour convertir les dates en format datetime
def convert_to_datetime(x):
    if isinstance(x, str):
        try:
            return datetime.strptime(x, '%Y%m%d')
        except ValueError:
            return np.nan
    return x

# Appliquer la fonction à la colonne 'FER_DAT_EN'
dh['FER_DAT_EN'] = dh['FER_DAT_EN'].apply(convert_to_datetime)
dh['FER_DT_SOR'] = dh['FER_DT_SOR'].apply(convert_to_datetime)
dh['FER_DAT_NA'] = dh['FER_DAT_NA'].apply(convert_to_datetime)
# Vérifier le résultat
print(dh)
```

Figure 3.7: Code de data time

La modification entraîne la nécessité d'une nouvelle vérification des types de données, voir la figure 3.8.

```
df.dtypes.value_counts()
```

```
object            32
float64           14
datetime64[ns]     5
int64              3
dtype: int64
```

Figure 3.8: Les types des données de df

Ce code affiche les types de données: dans le DataFrame, on trouve 32 chaînes de caractères, 14 nombres à virgule flottante, 5 colonnes représentant des dates et heures avec une précision à la nanoseconde, et 3 colonnes contenant des nombres entiers. Pour une meilleure visualisation, ces informations peuvent être représentées sous forme de diagramme circulaire, voir figure 3.9.

```
df.dtypes .value_counts().plot.pie()
```

```
<Axes: >
```

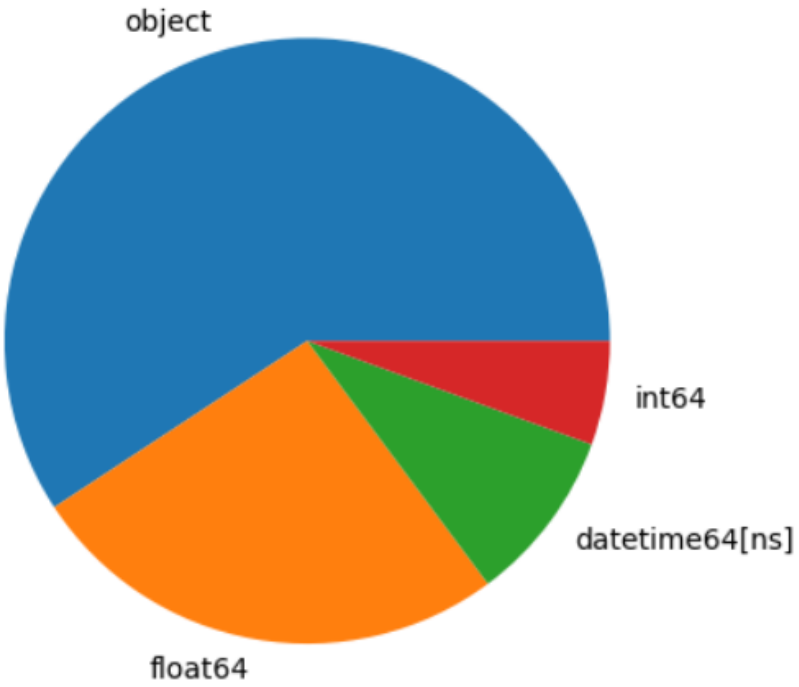


Figure 3.9: Diagramme circulaire pour les types de données

À partir de ce diagramme circulaire, il est possible d'observer que la base de données contient principalement des chaînes de caractères. Le diagramme indique en détail : 32 objets, 14 nombres à virgule flottante (float64), 5 données de type datetime64[ns], et 3 entiers (int64). Suite à l'analyse des données, la recherche et la visualisation des valeurs manquantes permettent de mieux comprendre leur répartition.

```
df.isna()
```

Figure 3.10: Code pour les valeurs nulles de df

```
: plt.figure(figsize=(20 ,10) )
  sns.heatmap(df.isna(), cbar= False)
```

Figure 3.11: Script seaborn pour les valeurs manquantes de df

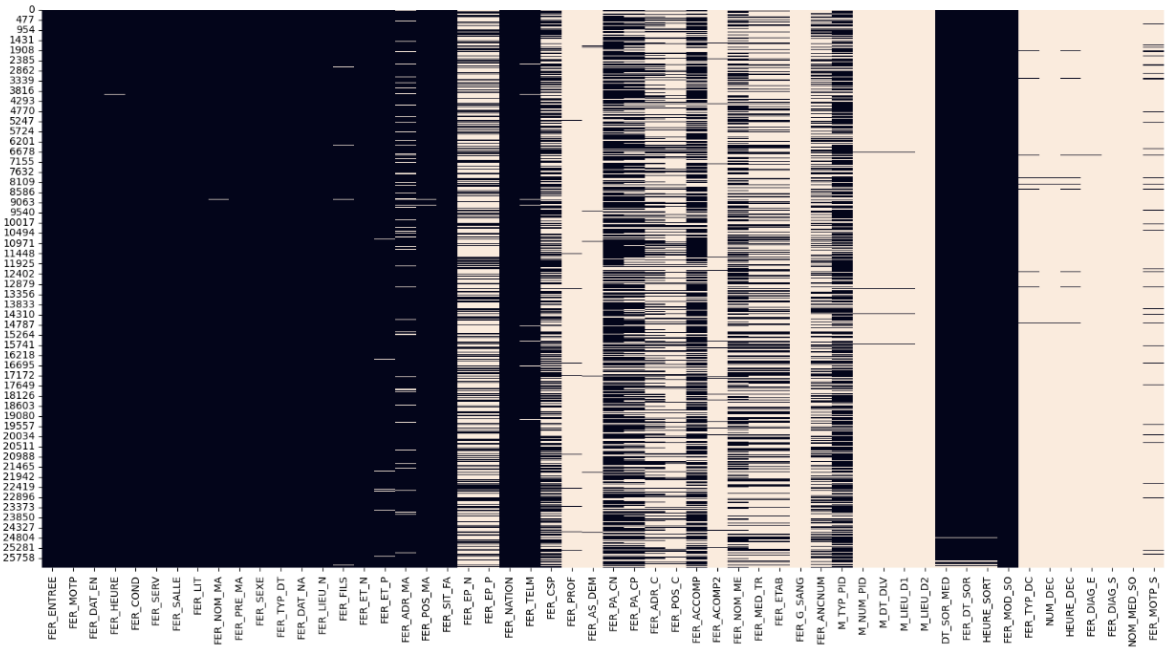


Figure 3.12: Un graphique en carte de chaleur

L'analyse des données révèle un grand nombre de valeurs manquantes nécessitant suppression.
Il est très important de connaître le pourcentage des valeurs manquantes, voici les resultats :

```
FER_ENTREE     0.000000
FER_NATION     0.000000
FER_DAT_EN     0.000000
FER_COND       0.000000
FER_SEXE       0.000000
FER_LIT        0.000000
FER_DAT_NA     0.000038
FER_SIT_FA     0.000038
FER_HEURE      0.000038
FER_SERV       0.000076
FER_SALLE      0.000076
FER_TYP_DT     0.000115
FER_MOTP       0.000153
FER_NOM_MA     0.000344
FER_PRE_MA     0.000764
FER_LIEU_N     0.000916
FER_MOD_SO     0.001871
FER_ET_N       0.002863
FER_POS_MA     0.005192
FER_FILS       0.005421
FER_ET_P       0.008170
FER_TELM       0.009124
DT_SOR_MED     0.013934
HEURE_SORT     0.013934
FER_DT_SOR     0.013934
FER_ADR_MA     0.089753
M_TYP_PID      0.259220
FER_PA_CN      0.262808
FER_ACCOMP     0.318661
FER_PA_CP      0.380125
```

Figure 3.13: Pourcentage des valeurs manquantes (1)

```
FER_CSP        0.449263
FER_NOM_ME     0.484958
FER_EP_N       0.617202
FER_EP_P       0.617202
FER_ANCNUM     0.634878
FER_ADR_C      0.649805
FER_MED_TR     0.668130
FER_ETAB       0.668741
FER_POS_C      0.790792
FER_MOTP_S     0.944567
FER_ACOMP2     0.976750
FER_PROF       0.980873
M_DT_DLV       0.989730
M_LIEU_D1      0.989730
M_NUM_PID      0.989730
FER_TYP_DC     0.991983
HEURE_DEC      0.991983
FER_AS_DEM     0.992097
NUM_DEC        0.994693
FER_G_SANG     0.999122
FER_DIAG_E     0.999275
M_LIEU_D2      1.000000
NOM_MED_SO     1.000000
FER_DIAG_S     1.000000
dtype: float64
```

Figure 3.14: Pourcentage des valeurs manquantes (2)

```
# Vérifier les valeurs manquantes dans dh
missing_values = dh.isnull().sum()
# Calculer le pourcentage de valeurs manquantes
missing_percentage = missing_values / len(dh) * 100

# Afficher les colonnes avec plus de 80% de valeurs manquantes
columns_to_drop = missing_percentage[missing_percentage > 80].index
print("Colonnes à supprimer en raison de trop de valeurs manquantes :")
print(columns_to_drop)
```

```
Colonnes à supprimer en raison de trop de valeurs manquantes :
Index(['FER_PROF', 'FER_AS_DEM', 'FER_ACOMP2', 'FER_G_SANG', 'M_NUM_PID',
       'M_DT_DLV', 'M_LIEU_D1', 'M_LIEU_D2', 'FER_TYP_DC', 'NUM_DEC',
       'HEURE_DEC', 'FER_DIAG_E', 'FER_DIAG_S', 'NOM_MED_SO', 'FER_MOTP_S'],
      dtype='object')
```

Figure 3.15: Scripte vérifie les valeurs manquantes et affiche les colonnes plus de 80 %

L'analyse des valeurs manquantes dans la base de données indique que certaines colonnes présentent un pourcentage élevé de valeurs manquantes, supérieur à 80%. Il est donc conseillé de les exclure de l'analyse ou du modèle afin d'éviter toute distorsion ou imprécision dans les conclusions.

```
# Supprimer les colonnes avec plus de 80% de valeurs manquantes
dh.drop(columns=columns_to_drop, inplace=True)

# Vérifier les données après suppression des colonnes
print(dh.head())
```

Figure 3.16: Script supprime et verfie les données supprimeés

Les colonnes présentant plus de 80% de valeurs manquantes sont supprimées. Ensuite, les données restantes sont vérifiées pour assurer leur intégrité.

```
          FER_HEURE      FER_COND      FER_SERV       FER_CSP
count  14129.000000  14129.000000  14129.000000  14129.000000
mean    1268.292377      2.799278    200.032557      6.428197
std      581.694812      1.464876     48.193343      5.808862
min        0.000000      1.000000    112.000000      0.000000
25%      847.000000      1.000000    163.000000      1.000000
50%     1216.000000      4.000000    232.000000      4.000000
75%     1728.000000      4.000000    232.000000      9.000000
max     2359.000000      4.000000    232.000000     65.000000
       FER_MOTP FER_SEXE  FER_ET_N FER_SIT_FA FER_NATION FER_DATE_EN  \
count     14129    14129     14129      14129      14129       14129
unique        1        4      5333          4          7         365
top      PATIEN        2  YAHIAOUI          M         DZ  2023-03-01
freq      14129     8329       105       8127      14120          66

       FER_TIME_EN
count        14129
unique           1
top       00:00:00
freq         14129
```

Figure 3.17: Affichage de la vérification

Suite à la suppression des colonnes contenant plus de 80% de valeurs manquantes, le nombre d'enregistrements pour chaque colonne restante est maintenant de 14 129. Les valeurs varient pour les heures et les services, et des valeurs spécifiques et fréquentes apparaissent pour les conditions, les professions, les noms, les sexes et les dates d'entrée.

```
# Distribution des motifs
print(dt['FER_MOTP'].value_counts())
```

```
PATIEN    14129
Name: FER_MOTP, dtype: int64
```

Figure 3.18: Distribution des motifs

La distribution révèle des valeurs répétitives, en particulier pour FER_MOTP, où une seule valeur se répète fréquemment (chaîne de caractères : 'patient').
L'exploitation optimale des données est une préoccupation constante pour un data scientist. Pour approfondir la compréhension des données, diverses techniques peuvent être employées. Voici quelques exemples.

```
# Nombre d'entrées par jour
print(dt['FER_DATE_EN'].value_counts().sort_index())
```

```
2022-12-20     1
2023-01-01    38
2023-01-02    55
2023-01-03    33
2023-01-04    34
              ..
2023-12-26    48
2023-12-27    55
2023-12-28    28
2023-12-29     3
2023-12-30     4
Name: FER_DATE_EN, Length: 365, dtype: int64
```

Figure 3.19: Nombre d'entrée par jour

L'image précédente illustre clairement le nombre d'entrées par jour, ce qui peut m'aider à analyser mes données.

```
# Calcul de la moyenne
entries_per_day = dt['FER_DATE_EN'].value_counts().sort_index()
average_entries_per_day = entries_per_day.mean()
print("Moyenne du nombre d'entrées par jour :", average_entries_per_day)
```

```
Moyenne du nombre d'entrées par jour : 38.70958904109589
```

Figure 3.20: Calculer la moyenne d'entrée

Le nombre d'entrées par jour est initialisé dans la variable `entries_per_day`. La fonction `mean` calcule que la moyenne du nombre d'entrées par jour pour l'année 2023 est d'environ 38 à 39 entrées.

```
# Calcul de la durée de séjour
dt['DURATION'] = (dt['FER_DT_SOR'] - dt['FER_DATE_EN']).dt.days
```

Figure 3.21: Script calcule la durée de séjour

```
# Moyenne de la durée de séjour
print(dt['DURATION'].mean())
```

```
1.9835798711869206
```

Figure 3.22: La moyenne de la durée de sejour

L'absence de la durée de séjour des patients dans le DataFrame représente un élément essentiel manquant pour la prédiction. La durée de séjour a été calculée en soustrayant la date de sortie de la date d'entrée, puis la fonction 'mean' a permis d'obtenir la moyenne de cette durée, qui varie entre une journée et deux jours. Pour une meilleure compréhension des données, l'utilisation de graphiques s'avère utile. Quelques exemples de ces graphiques permettent d'illustrer les résultats de manière plus claire et visuelle.

```
# Ligne du temps des entrées
df.set_index('FER_DAT_EN')['FER_ENTREE'].resample('D').count().plot(title='Nombre d\'entrées par jour')
plt.xlabel('Date')
plt.ylabel('Nombre d\'entrées')
plt.show()
```

Figure 3.23: Script affiche graphe date et nombre d'entrée

Ce code crée un graphique en ligne indiquant le nombre d'entrées journalières à l'hôpital. En utilisant les dates d'entrée comme index et en comptant les entrées par jour, il permet de visualiser les tendances et les variations quotidiennes dans les admis.

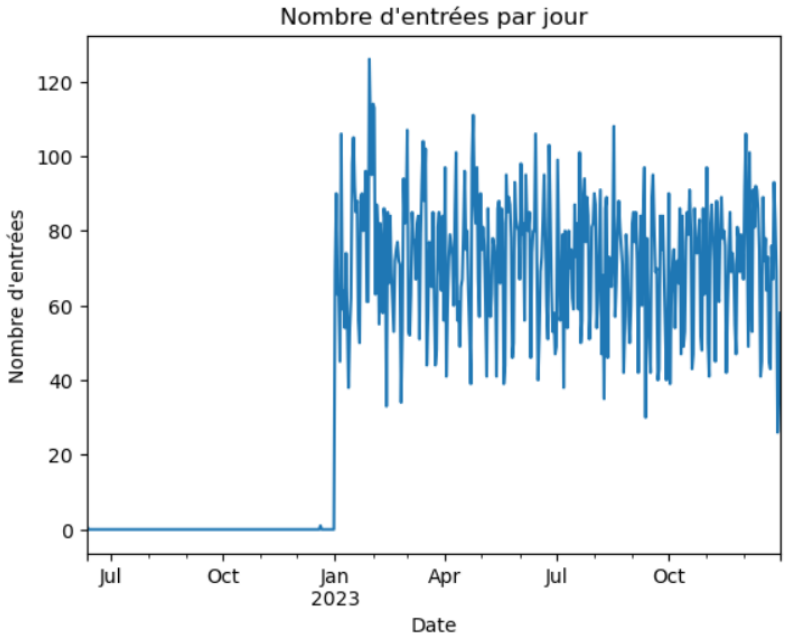


Figure 3.24: Graphe visualise nombre d'entrée par jour

Le graphique montre un motif de fluctuations régulières dans le nombre d'entrées journalières, ressemblant à un battement de cœur, de début 2023 à fin 2023. Cela suggère des cycles hebdomadaires, indiquant des jours de la semaine avec plus ou moins d'entrées, il pourrait y avoir plus d'entrées en début de semaine et moins en fin de semaine. ce qui peut aider à optimiser la gestion des ressources hospitalières.

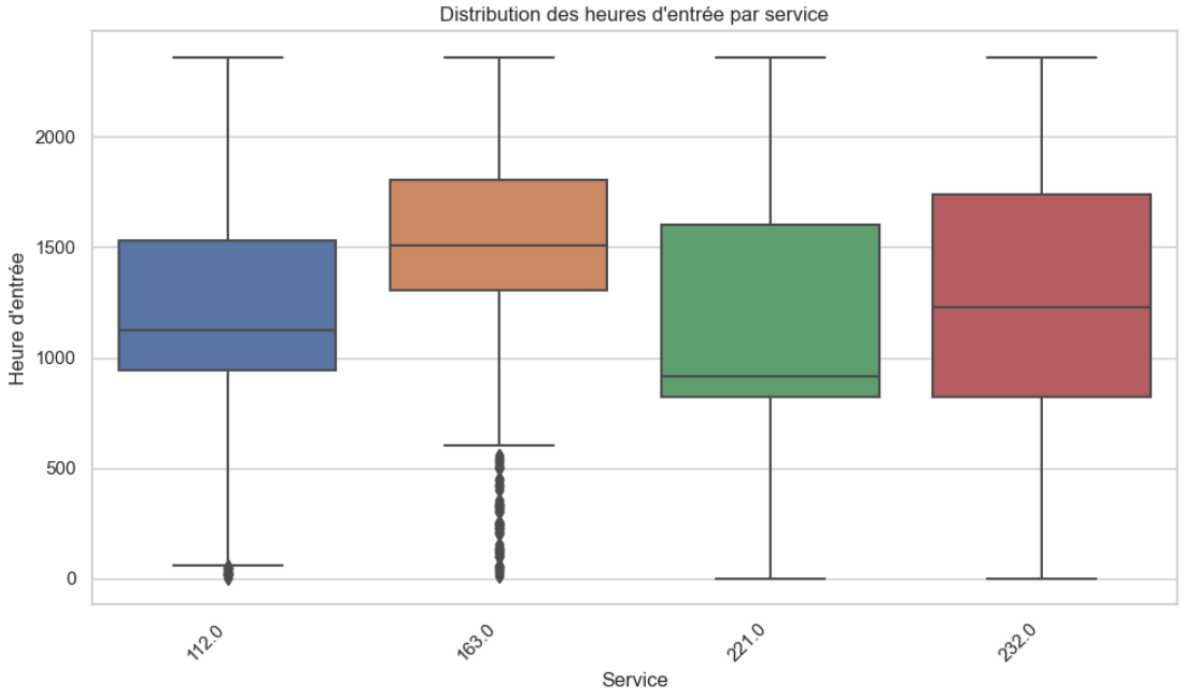


Figure 3.25: Graphe boite a moustache

Pour bien comprendre le graphique, il est important de savoir que chaque chiffre représente un service médical : 112 pour la médecine infantile, 163 pour la néonatologie, 221 pour la chirurgie infantile, et 232 pour la gynécologie obstétrique.

Boîte : Représente les valeurs interquartiles (du premier quartile Q_1 au troisième quartile Q_3).

Moustaches : Indiquent l'étendue des données, allant généralement de $Q_1 - 1.5 \times IQR$ à $Q_3 + 1.5 \times IQR$, où IQR est l'intervalle interquartile.

Médiane : La ligne à l'intérieur de la boîte qui représente la médiane des données.

Points Extérieurs : Points qui se situent au-delà des moustaches, souvent appelés valeurs atypiques ou outliers.

Interprétation Spécifique de la Boîte à Moustaches

Étapes d'Interprétation

Identifier les Quartiles :

- **Q1 (Premier Quartile) :** 25% des données se situent en dessous de cette valeur.
- **Q2 (Médiane) :** 50% des données se situent en dessous de cette valeur.
- **Q3 (Troisième Quartile) :** 75% des données se situent en dessous de cette valeur.

Identifier l'Étendue des Moustaches :

- Les moustaches montrent l'étendue des valeurs qui ne sont pas considérées comme des outliers.

Identifier les Outliers :

- Les points situés en dehors des moustaches représentent des valeurs atypiques.

```python
from sklearn.cluster import KMeans


# Assurez-vous que data_h_cleaned contient les colonnes numériques sans valeurs manquantes
numeric_cols = dt.select_dtypes(include=['number'])

# Exécutez KMeans clustering
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
dt.loc[:, 'Cluster'] = kmeans.fit_predict(numeric_cols)

# Visualisation des clusters
sns.scatterplot(x='FER_HEURE', y='FER_COND', hue='Cluster', data=dt)
plt.title('Clustering des patients')
plt.show()
```

Figure 3.26: Script cluster

Dans cette analyse, l'utilisation de la bibliothèque sklearn.cluster a permis d'importer et d'appliquer l'algorithme de clustering KMeans sur le jeu de données, segmentant ainsi les patients en différents groupes basés sur des caractéristiques similaires. Les colonnes numériques

pertinentes ont été sélectionnées pour exécuter le clustering avec trois clusters. Chaque patient a ensuite été assigné à un cluster spécifique selon les similarités présentes dans leurs données. Un scatter plot a été utilisé pour visualiser la répartition des clusters en fonction des heures d'admission (FER_HEURE) et de l'état du patient à l'entrée (FER_COND). Les points du graphique, colorés selon leur cluster, ont permis d'observer distinctement les différents groupes formés.

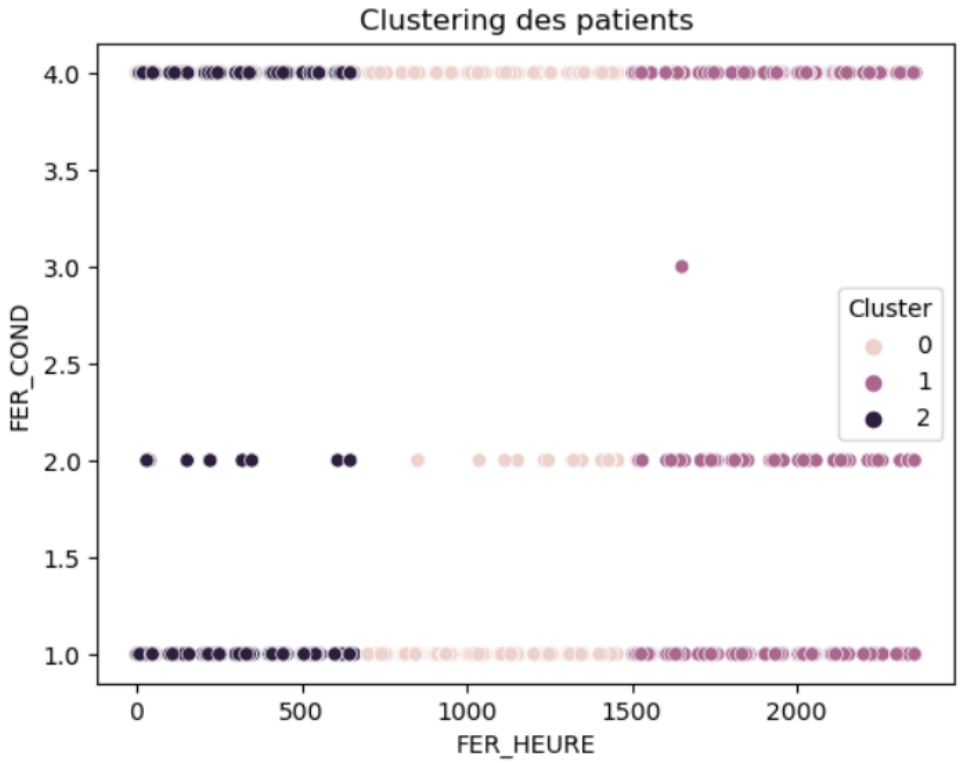


Figure 3.27: Graphe de clustering

À partir du graphique de clustering des patients, trois clusters distincts ont été identifiés : 4, 2 et 1, correspondant aux classes 0, 1 et 2 . Regroupant les patients partageant des caractéristiques similaires. En observant les heures d'admission (FER_HEURE), celles-ci apparaissent relativement équilibrées entre les clusters, sans concentration notable à des moments spécifiques de la journée. Cela suggère que les admissions des patients se répartissent de manière assez uniforme. Les clusters sont bien définis et séparés, indiquant une bonne performance du modèle de clustering pour regrouper les patients en fonction de leurs caractéristiques. Ces clusters distincts permettent de capturer des groupes de patients ayant des profils différents, notamment en termes d'heures d'admission et de l'état du patient à l'entrée (FER_COND).

```
# Convertir la colonne Duree_sejour en valeurs numériques représentant le nombre de jours
dt['Duree_sejour_numerique'] = dt['Duree_sejour'].dt.days
```

Figure 3.28: Code pour convertir en nombre numérique

Conversion de la colonne 'Duree_sejour' en valeurs numériques représentant le nombre de

jours pour faciliter l'analyse quantitative des séjours des patients.

```python
def remove_outliers_iqr_all(dt):
    df_filtered = dt.copy()
    removed_indices = []  # Pour stocker les indices des lignes supprimées
    for column in dt.select_dtypes(include=['float64', 'int64']).columns:
        Q1 = df_filtered[column].quantile(0.25)
        Q3 = df_filtered[column].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        outliers_mask = (df_filtered[column] < lower_bound) | (df_filtered[column] > upper_bound)
        removed_indices.extend(df_filtered.index[outliers_mask].tolist())
        df_filtered = df_filtered[~outliers_mask]
    removed_indices = list(set(removed_indices))  # Supprimer les doublons des indices
    removed_values = dt.loc[removed_indices]  # Récupérer les valeurs supprimées
    return df_filtered, removed_values

# Appliquer la fonction remove_outliers_iqr_all pour supprimer les valeurs aberrantes de toutes les colonnes numériques dans dt
data_fi, removed_values = remove_outliers_iqr_all(dt)

# Afficher le nombre de lignes avant et après suppression des valeurs aberrantes
print("Nombre de lignes avant suppression des valeurs aberrantes :", len(dt))
print("Nombre de lignes après suppression des valeurs aberrantes :", len(data_fi))
```

```
Nombre de lignes avant suppression des valeurs aberrantes : 14129
Nombre de lignes après suppression des valeurs aberrantes : 8709
```

Figure 3.29: Fonction pour supprimer les valeurs aberrantes

Utilisation d'une fonction pour supprimer les valeurs aberrantes dans toutes les colonnes numériques du DataFrame via la méthode de l'écart interquartile (IQR). Détection et élimination des valeurs situées en dehors des bornes définies par 1,5 fois l'IQR, permettant de nettoyer les données en supprimant les points extrêmes. Amélioration potentielle de la qualité des analyses et des modèles. Après application de la fonction, réduction du nombre de lignes de 14 129 à 8 709, indiquant une élimination significative des points extrêmes.

```python
# Supprimer les lignes où Duree_sejour_numerique est égal à 0
data_fi = data_fi[data_fi['Duree_sejour_numerique'] != 0]
```

Figure 3.30: Code pour supprimer les lignes égales à zéro

Suppression des lignes du DataFrame où la durée de séjour numérique (Duree_sejour_numerique) est égale à 0, permettant ainsi un nettoyage plus approfondi des données en éliminant les séjours de durée nulle susceptibles de fausser les analyses.

```python
# Nombre de lignes et de collonne  contenant des valeurs manquantes
nombre_collonnes_valeurs_manquantes = data_fi.isnull().any(axis=0).sum()
nombre_lignes_valeurs_manquantes = data_fi.isnull().any(axis=1).sum()
print(f"Nombre de collonnes  avec des valeurs manquantes : {nombre_collonnes_valeurs_manquantes}")
print(f"Nombre de lignes avec des valeurs manquantes : {nombre_lignes_valeurs_manquantes}")
```

```
Nombre de collonnes  avec des valeurs manquantes : 0
Nombre de lignes avec des valeurs manquantes : 0
```

Figure 3.31: Affichage du nombre de lignes et de colonnes égales à zéro

Vérification des valeurs manquantes dans le DataFrame nettoyé data_fi, montrant l'absence de colonnes ou de lignes contenant des valeurs manquantes. Les données sont donc complètes

et prêtes pour une analyse ultérieure sans nécessiter de gestion supplémentaire des valeurs manquantes.

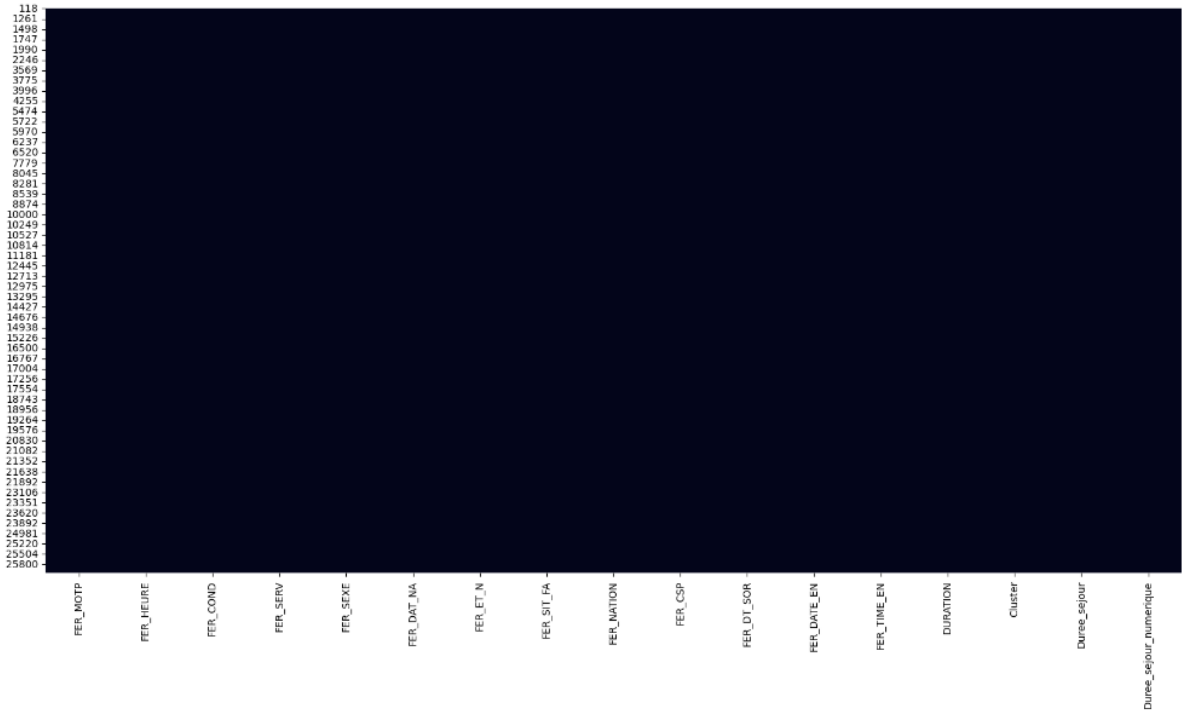


Figure 3.32: Carte de Chaleur des Valeurs Manquantes

Observation directe à l'aide de cette image Seaborn montrant l'absence de valeurs manquantes par rapport au premier graphique.

Après avoir complété les étapes de nettoyage de données, le DataFrame data_fi est désormais prêt pour l'analyse .

3.2.4 Formulation des hypothèses

Après le nettoyage et l'exploration des données, la formulation des hypothèses suit. Plusieurs hypothèses sont posées pour guider l'étude sur la prédiction de la durée de séjour des patients. Il est supposé que des caractéristiques telles que le sexe, l'état de santé à l'admission, le service de traitement et le cluster de patients influencent significativement la durée de séjour à l'hôpital. L'utilisation de ces variables dans le modèle de prédiction est prévue pour fournir des résultats précis et significatifs.

3.2.5 Détermination des variables synthétiques

Pour réaliser la prédiction, certaines caractéristiques, dont l'âge, sont nécessaires. Cependant, l'âge n'est pas disponible dans le DataFrame actuel. Il doit donc être créé à partir de la date de naissance.

```
# Calculer l'âge à partir de la date de naissance
data_fi['FER_AGE'] = pd.to_datetime('today').year - pd.to_datetime(data_fi['FER_DAT_NA']).dt.year
```

Figure 3.33: Calculer l'age

Ce code calcule l'âge des individus à partir de leur date de naissance et ajoute cette information au DataFrame data_fi dans une nouvelle colonne appelée FER_AGE. Il convertit d'abord la colonne des dates de naissance en format datetime, puis extrait l'année de naissance. En soustrayant cette année de l'année actuelle, il obtient l'âge des individus, qu'il stocke dans la nouvelle colonne FER_AGE du DataFrame.

Détermination des variables synthétiques parallèlement au nettoyage des données et à la formulation des hypothèses pour capturer plus précisément les caractéristiques des patients et enrichir l'analyse. Création de la variable 'Duree_sejour_numerique' pendant le nettoyage, calculée comme la différence entre les dates d'entrée et de sortie des patients, permettant de quantifier la durée de séjour de chaque patient. Application d'une méthode de clustering lors de l'exploration initiale des données pour regrouper les patients selon leurs caractéristiques similaires, résultant en la variable 'Cluster'. Cette variable s'est révélée particulièrement utile pour visualiser les groupes de patients et leurs comportements au sein de l'hôpital. Ces variables synthétiques ont joué un rôle crucial dans le test des hypothèses et l'amélioration des performances des modèles prédictifs.

3.2.6 Construction du modèle

À l'étape de construction du modèle pour prédire la durée de séjour des patients à l'hôpital à l'aide d'une technique de régression avancée, il est crucial de définir au préalable la variable cible ainsi que les caractéristiques (features).

```
# Sélectionner les features et la cible
features = ['FER_HEURE', 'FER_COND', 'FER_SERV', 'FER_CSP', 'Cluster', 'FER_SEXE','FER_AGE']
target = 'Duree_sejour_numerique'
```

Figure 3.34: Séléctioner les features et la cible

Ces lignes de code définissent les variables features et target pour un modèle. Les features sont les caractéristiques que le modèle utilisera pour faire des prédictions, telles que l'heure d'entrée (FER_HEURE), la condition du patient à l'entrée (FER_COND), le service (FER_SERV), la catégorie socio-professionnelle (FER_CSP), le cluster, le sexe (FER_SEXE), et l'âge (FER_AGE). La variable target est la variable que le modèle tentera de prédire, ici la durée de séjour numérique (Duree_sejour_numerique). Après avoir sélectionné la variable cible et les caractéristiques, le choix de l'algorithme pour prédire la durée de séjour est nécessaire. L'algorithme Random Forest a été retenu pour cette tâche.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_scor
```

Figure 3.35: Importer des bib Sklearn

Utilisation de la bibliothèque sklearn.ensemble pour importer un modèle de machine learning Random Forest., et les bibliothèques sklearn.metrics et sklearn.model_selection pour importer les fonctions MSE, RMSE et R², afin d'évaluer mon modèle et déterminer sa performance.

```
# Séparation des données
X = data_fi[features]
y = data_fi[target]

# Créer un modèle Random Forest Regression
rf_model = RandomForestRegressor(random_state=42, n_estimators=200)

# Évaluation avec la validation croisée
cv_scores = cross_val_score(rf_model, X, y, cv=5, scoring='neg_mean_squared_error')
mse_mean = -cv_scores.mean()
rmse_mean = mse_mean ** 0.5

# Entraînement du modèle
rf_model.fit(X, y)

# Prédiction
y_pred = rf_model.predict(X)
```

Figure 3.36: Script algo RF

Ce script effectue plusieurs étapes pour construire un modèle d'apprentissage automatique, en l'occurrence un Random Forest. Tout d'abord, il sépare les données en caractéristiques (X) et en valeurs cibles (y). Ensuite, il crée un modèle Random Forest avec 200 arbres de décision, en utilisant une seed de 42 pour assurer la reproductibilité des résultats. Une fois cela fait, l'étape cruciale de l'évaluation croisée intervient pour valider la performance du modèle. Ensuite, le modèle est entraîné sur les données d'entraînement, suivies par la prédiction des valeurs cibles, qui sont stockées dans y_pred.

```
# Calculer le MSE et le RMSE sur les données d'entraînement
mse = mean_squared_error(y, y_pred)
rmse = mse ** 0.5
# Calculer le R²
r2 = r2_score(y, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
print(f'R²: {r2}')
```

Figure 3.37: Script pour afficher les perfotmances

Ce script évalue les performances du modèle Random Forest sur les données d'entraînement en calculant plusieurs métriques clés. Tout d'abord, le **Mean Squared Error (MSE)** est calculé pour mesurer l'erreur quadratique moyenne entre les valeurs réelles et les valeurs prédites, fournissant une indication de la précision globale du modèle. Ensuite, le **Root Mean Squared Error (RMSE)**, qui est la racine carrée du MSE, est déterminé pour exprimer l'erreur moyenne dans les mêmes unités que les valeurs cibles, facilitant ainsi l'interprétation. Enfin, le **Coefficient de Détermination** (R^2) est calculé pour évaluer la proportion de la variance des valeurs réelles qui est expliquée par le modèle. Ces métriques sont ensuite affichées, offrant une vision claire et quantitative de la performance du modèle Random Forest.

```
Mean Squared Error: 1.3716705043552913
Root Mean Squared Error: 1.1711833777659633
R²: 0.8292625709331352
```

Figure 3.38: Les métriques de performance

Les valeurs obtenues pour les métriques de performance sont les suivantes : un MSE de 1,73 et un RMSE de 1,17. Plus ces valeurs sont basses, plus le modèle est précis. Le coefficient de détermination (R^2) est de 0,8293 ce qui signifie que 82.93% de la variance des valeurs réelles est expliquée par le modèle. Un R^2 proche de 1 indique que le modèle a une bonne capacité prédictive. En résumé, ces métriques montrent que mon modèle a une performance solide sur les données d'entraînement.

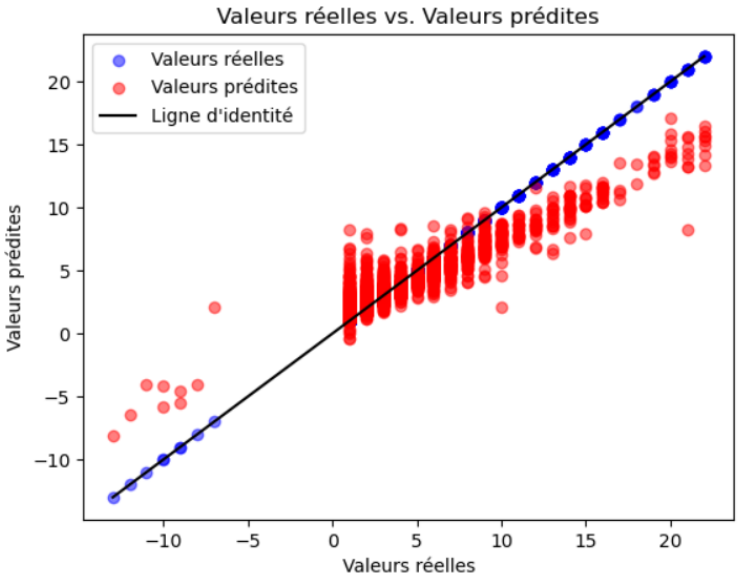


Figure 3.39: Graphe de comparaison

Pour bien comprendre mes valeurs, une visualisation est nécessaire. Ce graphique compare les valeurs réelles aux valeurs prédites. La ligne noire représente la ligne d'identité, et les valeurs réelles sont proches des valeurs prédites, à l'exception de quelques points éloignés qui constituent des erreurs mineures. Globalement, le modèle semble performant pour la majorité des données, particulièrement dans la plage de 0 à 10 pour les valeurs réelles.

```
print(y.mean())
print(y_pred.mean())
```

```
2.6153713892709765
2.5885014259549024
```

Figure 3.40: Comparé le moyennes

La similarité entre la moyenne des valeurs réelles et celles prédites indique que les prédictions sont proches de la réalité, ce qui montre que le modèle fonctionne bien.

Comparaison

```python
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score

# Sélectionner les features et la cible
features = ['FER_HEURE', 'FER_COND', 'FER_SERV', 'FER_CSP', 'Cluster', 'FER_SEXE', 'FER_AGE']
target = 'Duree_sejour_numerique'

# Séparation des données
X = data_fi[features]
y = data_fi[target]

# Créer un modèle Gradient Boosting Regression
gb_model = GradientBoostingRegressor(random_state=42, n_estimators=200)

# Entraînement du modèle
gb_model.fit(X, y)
# Prédiction
y_pred = gb_model.predict(X)
# Calculer le MSE et le RMSE sur les données d'entraînement (ceci n'est pas recommandé en pratique,
# mais nous le faisons ici à des fins de démonstration)
mse = mean_squared_error(y, y_pred)
rmse = mse ** 0.5
# Calculer le R²
r2 = r2_score(y, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
print(f'R²: {r2}')
```

```
Mean Squared Error: 6.052932415272613
Root Mean Squared Error: 2.460270801207179
R²: 0.2465667843569478
```

Figure 3.41: Script de modèle Gradient Boosting Regression

Ce code utilise l'algorithme de Gradient Boosting Regression pour prédire la durée de séjour des patients en se basant sur plusieurs caractéristiques telles que l'heure d'entrée, la condition du patient, le service hospitalier, la catégorie socioprofessionnelle, le cluster auquel appartient le patient, son sexe et son âge. Le modèle est entraîné sur ces données et ensuite utilisé pour faire des prédictions. Ensuite, il calcule des métriques comme le Mean Squared Error (MSE), le Root Mean Squared Error (RMSE) et le R-squared (R^2) pour évaluer la précision du modèle. Ces métriques permettent de mesurer à quel point les prédictions du modèle correspondent aux valeurs réelles de la durée de séjour, ce qui est crucial pour évaluer son efficacité.

Pour les résultats obtenus, le modèle de Gradient Boosting Regression (GBR) affiche un MSE de 0.65, un RMSE de 2.46 et un R^2 de 0.25. Ces valeurs indiquent que le modèle n'est pas très fiable.

```python
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score

# Sélectionner les features et la cible
features = ['FER_HEURE', 'FER_COND', 'FER_SERV', 'FER_CSP', 'Cluster', 'FER_SEXE', 'FER_AGE']
target = 'Duree_sejour_numerique'

# Séparation des données
X = data_fi[features]
y = data_fi[target]

# Créer un modèle Régression Ridge
ridge_model = Ridge(alpha=1.0, random_state=42)

# Entraînement du modèle
ridge_model.fit(X, y)

# Prédiction
y_preda = ridge_model.predict(X)

# Calculer le MSE et le RMSE sur les données d'entraînement (ceci n'est pas recommandé en pratique,
# mais nous le faisons ici à des fins de démonstration)
mse = mean_squared_error(y, y_pred)
rmse = mse ** 0.5

# Calculer le R²
r2 = r2_score(y, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
print(f'R²: {r2}')
```

```
Mean Squared Error: 7.2624718641392505
Root Mean Squared Error: 2.6948973754373746
R²: 0.09601046984939976
```

Figure 3.42: Scripte de modèle Ridge

Ce code utilise la régression Ridge pour construire un modèle prédictif. Il commence par importer les bibliothèques nécessaires et sélectionne les caractéristiques (features) ainsi que la variable cible à partir des données. Ensuite, il divise les données en variables explicatives (X) et variable cible (y). Le modèle de régression Ridge est créé avec un paramètre alpha par défaut de 1.0 pour la régularisation, puis il est entraîné sur les données d'entraînement. Les valeurs cibles sont prédites à partir des données d'entraînement, et ensuite le MSE, le RMSE et le R^2 sont calculés pour évaluer la performance du modèle. Enfin, les résultats sont affichés pour fournir un aperçu de la qualité de l'ajustement du modèle.

Concernant les résultats de l'évaluation de la régression Ridge, on constate que le modèle n'a pas une performance optimale sur les données d'entraînement. Le MSE (Mean Squared Error) élevé de 7.26 et le RMSE (Root Mean Squared Error) de 2.69 révèlent des écarts significatifs entre les valeurs réelles et prédites. De plus, le coefficient de détermination R^2 de 0.10 indique que seulement environ 10% de la variance des valeurs cibles est expliquée par le modèle.

Table 3.1: Comparaison des Algorithmes de prédiction

Algorithme	**MSE**	**RMSE**	**R²**
Random Forest	1.372	1.171	0.829
Ridge	7.262	2.695	0.096
Gradient Boosting Regressor	6.053	2.460	0.247

D'après les valeurs d'évaluation des modèles et la table de comparaison, il est clair que le modèle Random Forest présente des résultats significativement meilleurs. En effet, un MSE et un RMSE plus bas, ainsi qu'un R^2 plus élevé, indiquent une meilleure capacité du modèle Random Forest à expliquer la variance des valeurs cibles et à fournir des prédictions plus précises. C'est la raison pour laquelle j'ai choisi le modèle Random Forest comme le meilleur choix pour mon analyse.

Conclusion

Ce troisième chapitre a démontré le processus rigoureux de développement d'un modèle de machine learning, en suivant les étapes méthodologiques d'un data scientist. Après une analyse exploratoire et une préparation minutieuse des données, nous avons choisi un modèle adapté et l'avons entraîné en utilisant des techniques de validation croisée. Les résultats ont montré des performances prometteuses, avec une bonne capacité de prédiction et de généralisation.

Les prédictions du modèle, notamment sur les durées de séjour des patients, offrent des applications pratiques significatives, telles que l'optimisation des ressources et une meilleure planification des soins à la Maternité de Targa Ouzmour. Ce chapitre met en lumière l'importance d'une approche méthodique pour obtenir des résultats fiables et exploitables, ouvrant la voie à des améliorations futures dans la gestion des données et la prise de décision basée sur ces dernières.

4

Évaluation et Amélioration du Service d'Accueil : Critiques, Suggestions et Support Méthodologique

Introduction

Dans ce dernier chapitre, il sera question de la présentation des résultats, de la communication, des propositions de solutions aux problèmes rencontrés au bureau des entrées, ainsi que de l'introduction des outils utilisés lors de la réalisation du mémoire.

4.1 Présentation et communication

Après avoir terminé le modèle, la dernière étape consiste à présenter les résultats du travail.Voici le résultat:

	Valeurs Prédites
0	1.7
1	2.3
2	1.8
3	1.9
4	1.5
5	1.2
6	6.2
7	1.6
8	3.1
9	1.9
10	3.0
11	3.3
12	7.2
13	3.7
14	1.2
15	1.6
16	8.9
17	4.9
18	1.9
19	1.4

Figure 4.1: Le résulat de prédiction

Après avoir prédit la durée de séjour des patients, particulièrement dans les services mère et enfant, plusieurs actions peuvent être mises en œuvre pour améliorer la gestion et les soins au CHU de Béjaïa. Voici quelques suggestions concrètes :

1. **Optimisation de la gestion des lits :**

- **Planification des Admissions :** En connaissant la durée probable de séjour des patients, les admissions et les sorties peuvent être mieux planifiées, assurant ainsi une utilisation optimale des lits disponibles.
- **Réduction du Temps d'Attente :** Les prédictions permettent d'anticiper la disponibilité des lits, réduisant le temps d'attente des patients pour être admis, ce qui est particulièrement crucial dans les services de maternité et de pédiatrie.

2. **Allocation des ressources :**

   - **Personnel Médical :** Une meilleure prévision de la charge de travail permet d'optimiser la répartition du personnel médical, notamment des sages-femmes, pédiatres, et infirmières, pour répondre efficacement aux besoins des patients.
   - **Matériel Médical :** La gestion des équipements et des fournitures médicales peut être ajustée en fonction des prédictions, garantissant ainsi la disponibilité des ressources nécessaires, telles que les incubateurs pour les nouveau-nés.

3. **Amélioration des Soins aux Patients :**

   - **Personnalisation des Soins :** Les patients dont les séjours sont prédits comme étant plus longs peuvent bénéficier d'une attention particulière, ce qui améliore la qualité des soins pour les mères et les enfants.
   - **Suivi des Patients :** Une gestion proactive du suivi des patients, surtout ceux ayant des séjours prolongés, peut être mise en place pour assurer une continuité des soins après la sortie de l'hôpital.

4. **Planification Budgétaire:**

   - **Précision des Coûts:** En prévoyant la durée de séjour, les coûts associés aux soins des patients peuvent être estimés plus précisément, facilitant ainsi la planification budgétaire du service mère-enfant.
   - **Réduction des Coûts :** Une gestion plus efficace des séjours peut mener à des économies en réduisant les jours d'hospitalisation non nécessaires.

5. **Analyse et Amélioration Continue:**

   - **Identification des Tendances:** En analysant les prédictions de durée de séjour, des tendances et des facteurs influençant les durées de séjour peuvent être identifiés, permettant ainsi de mettre en place des mesures correctives adaptées.
   - **Évaluation des Performances :** Ces données peuvent également servir à évaluer les performances des différentes unités ou services et à identifier les domaines nécessitant des améliorations.

4.2 Les critiques

- Le matériel utilisé et les moyens informatiques ne sont pas de qualité satisfaisante.

- Les tarifs des consultations ne sont pas adaptés et certains cas bénéficient de consultations gratuites.
- L'espace de travail est restreint, ce qui rallonge considérablement la durée des tâches.
- Les patients attendent longtemps, ce qui prolonge la durée des consultations.
- Le logiciel de gestion des patients présente de nombreux inconvénients qui nécessitent une révision approfondie.
- Le bureau des entrées regroupe plusieurs services, tels que les statistiques , les naissance, les décès et les recouvrements. Cependant, le travail et les tâches sont anarchiquement répartis en raison de l'absence de bureaux dédiés à chaque service.
- La plupart des données sont stockées sous format papier.

4.3 Suggestions

- Acheter et renouveler du nouveau matériel , et Augmenter les ressources informatiques disponibles.
- Le problème identifié est que les tarifs des consultations ne sont pas adaptés, et certains cas bénéficient de consultations gratuites. La solution proposée est d'augmenter le tarif des quittances.
- Agrandir le bureau pour faciliter le travail des agents.
- Minimiser la durée d'attente des patients en utilisant des techniques d'optimisation.
- Développer un nouveau logiciel, appelé DEMDZ, qui surpasse le logiciel patient et évite ses inconvénients.
- Augmenter l'espace du bureau des entrées afin de fournir un bureau spécial pour chaque service.
- Toutes les données doivent être stockées au format numérique.

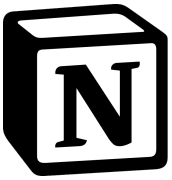
4.4 Outil de productivité et de communication

4.4.1 Notion

est un espace unique qui permet l'écriture et la planification. C'est une plateforme polyvalente qui permet la planification, la gestion de projets et même diriger une entreprise entière, Notion offre la flexibilité nécessaire pour organiser les idées et les informations de la manière qui convient le mieux à chaque équipe.[15]

J'ai utilisé Notion pour noter mes idées, mes interrogations et mes trouvailles. Il m'a également permis de stocker dans un endroit partagé la documentation des composants React.js que j'ai créés ainsi que les résumés et les bouts de code utiles des technologies que j'ai utilisées.

- Site officiel de Notion : https://www.notion.so



4.4.2 Discord

est une plateforme de communication en ligne qui permet aux utilisateurs de se connecter, de discuter et de collaborer via des appels vocaux, des appels vidéo, des messages texte et le partage de médias. La convivialité et la simplicité d'utilisation de Discord en font un outil populaire pour la communication en ligne.

- Site officiel de Discord : https://discord.com

4.5 Outil de conception et de prototypage

4.5.1 Figma

est une plateforme collaborative pour l'édition de graphiques vectoriels et le prototypage. Elle permet de créer des design système, facilite la conception de sites web et d'applications mobiles, et offre des fonctionnalités de design, de prototypage et de gestion de projet avancés. Figma est accessible via les navigateurs web, dispose d'une version bureau pour macOS et Windows, ainsi que d'une solution de visualisation des designs sur iOS et Android.[3]

- Site officiel de Figma https://figma.com

Conclusion

En conclusion , après avoir critiqué de manière constructive, les suggestions proposées visaient à améliorer la gestion générale rencontrée au bureau des entrées. Les outils présentés m'ont vraiment aidé à réaliser ce projet.

Conclusion générale et travaux futures

En résumé, cette étude a mis en évidence la nécessité d'améliorer la gestion des données au sein du bureau des entrées de la maternité de Targa Ouzmour. J'ai commencé par visualiser les données existantes pour mieux comprendre les flux et les processus actuels. Cette étape m'a permis d'identifier des inefficacités et des zones d'amélioration.

Utilisation de techniques de machine learning pour la prédiction de la durée de séjour des patients. Cette approche a montré un potentiel d'optimisation de la gestion des lits et des ressources, favorisant une meilleure planification et une réduction des temps d'attente.

Les résultats obtenus sont encourageants et démontrent que l'intégration de la science des données et du machine learning peut offrir des solutions concrètes et efficaces aux défis de la gestion hospitalière. Toutefois, certains objectifs n'ont pas été atteints, notamment la mise en place d'un portail patient permettant l'accès aux données médicales personnelles. Satisfaction quant aux progrès réalisés et aux perspectives ouvertes par cette étude, malgré certaines limitations. Défis restants à relever, notamment l'intégration de systèmes automatisés pour l'extraction des informations des cartes d'identité des patients, garantissant une collecte sécurisée et efficace des données essentielles.

Pour conclure, la modernisation de la gestion des données hospitalières est essentielle pour renforcer la qualité des soins et l'efficacité des services. Les outils et méthodes présentés dans cette étude établissent les fondements d'une gestion plus efficace et d'une amélioration continue des processus au sein des établissements de santé.

Engagement ferme à respecter la confidentialité des données, qu'elles soient personnelles ou non, dans ma future carrière professionnelle. Priorité accordée à la sécurité des informations, avec un traitement rigoureux des données conformément aux normes éthiques et légales les plus strictes. Conviction de l'importance de la protection de la vie privée des individus, en assurant l'accès aux informations uniquement aux personnes autorisées, tout en garantissant le respect des droits et de la confidentialité de chacun.

References

[1] Arbane, Z., and Ben Mohand Said, A. Le système d'information hospitalier, un préalable pour la mise en place d'un système d'information sanitaires : Cas du chu de tizi-ouzou. Master's thesis, Université Mouloud Mammeri de Tizi-Ouzou, 2023.

[2] Azencott, C.-A. *Introduction au Machine Learning*. Dunod, 2020.

[3] BDM. Figma - un outil de prototypage et de design collaboratif, 2023. Consulté le 10 juin 2024.

[4] Bouhissi, M. Module pour les data scientists, 2022/2023.

[5] Bouzidi, J. Les 7 étapes d'un projet data science. LinkedIn, 2019. 2024.

[6] CHU-Béjaïa. Site officiel du CHU de Béjaïa, rubrique Frantz Fanon, 2014. Consulté le 21 mars 2023.

[7] CHU-Béjaïa. Site officiel du chu de béjaïa, rubrique khellil amrane. https://www.chubejaia.dz/HopitalKhllil, 2014. Consulté le 21 mars 2023.

[8] CHU-Béjaïa. Site officiel du chu de béjaïa, rubrique targa ouzmour. https://www.chubejaia.dz/HopitalTarga, 2014. Consulté le 21 mars 2023.

[9] GeeksforGeeks. Ml classification vs regression, 2024. Consulté le 24 mai 2024.

[10] Géron, A. *Machine Learning avec Scikit-Learn*. O'Reilly Media, 2019. Traduit de l'anglais.

[11] Hassani, A. Anonymisation des données par l'apprentissage non supervisé. Mémoire de fin d'études, Universite´ Ahmed Draia - Adrar, Adrar, Algérie, 2022/2023. Option : Systèmes intelligents.

[12] Javapoint. Random forest algorithm, 2024. Visité le 05/2024.

[13] Matteis, L. D. *Introduction à l'apprentissage automatique*. éditeur, lieu de publication, année de publication.

[14] Mouhous, S., and Oulhaci, Y. *Conception et réalisation d'un système d'information hospitalier*. Université Mouloud Mammeri, 2022–2023.

[15] Notion. What is notion - guide, 2023. Consulté le 10 juin 2024.

[16] scikit-learn developers. *scikit-learn user guide*, June 2017. Release 0.18.2.

[17] Sublime, J. Titre de l'article. *Nom du journal* (2022), 12–34.

[18] Teston, R. Le machine learning au service de la santé. Consulté le 4 mars 2024.

[19] Vivek, J. La science des données dans le secteur de la santé : avantages, stratégies, applications, outils et tendances futures, 2024. Assistant Marketing Manager.

[20] Vivek, J. La science des données dans le secteur de la santé : avantages, stratégies, applications, outils et tendances futures, 2024. Assistant Marketing Manager.

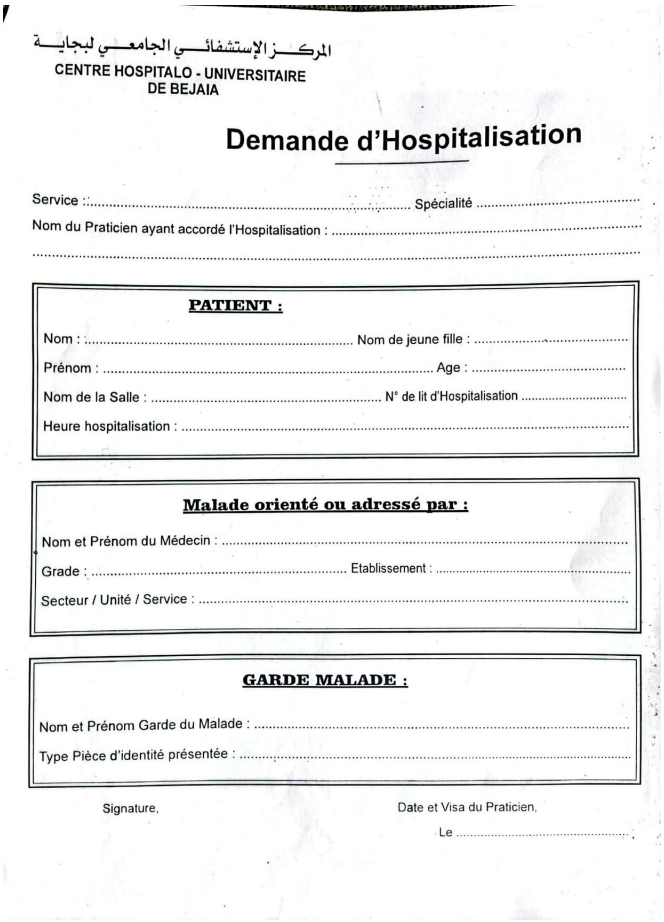
Annexes

4.6 Annexe 1

4.6.1 Liste des documents

Dans cette partie, je présente les documents principaux du dossier médical des patients.

La figure 4.2 ci-dessous représente le document qui permet au praticien de faire une demande d'hospitalisation.



المركـــز الإستشفائـــي الجامعـــي لبجايـــة
CENTRE HOSPITALO - UNIVERSITAIRE
DE BEJAIA

Demande d'Hospitalisation

Service :..... Spécialité .....
Nom du Praticien ayant accordé l'Hospitalisation : .....
.....

PATIENT :

Nom : ..... Nom de jeune fille : .....
Prénom : ..... Age : .....
Nom de la Salle : ..... N° de lit d'Hospitalisation .....
Heure hospitalisation : .....

Malade orienté ou adressé par :

Nom et Prénom du Médecin : .....
Grade : ..... Etablissement : .....
Secteur / Unité / Service : .....

GARDE MALADE :

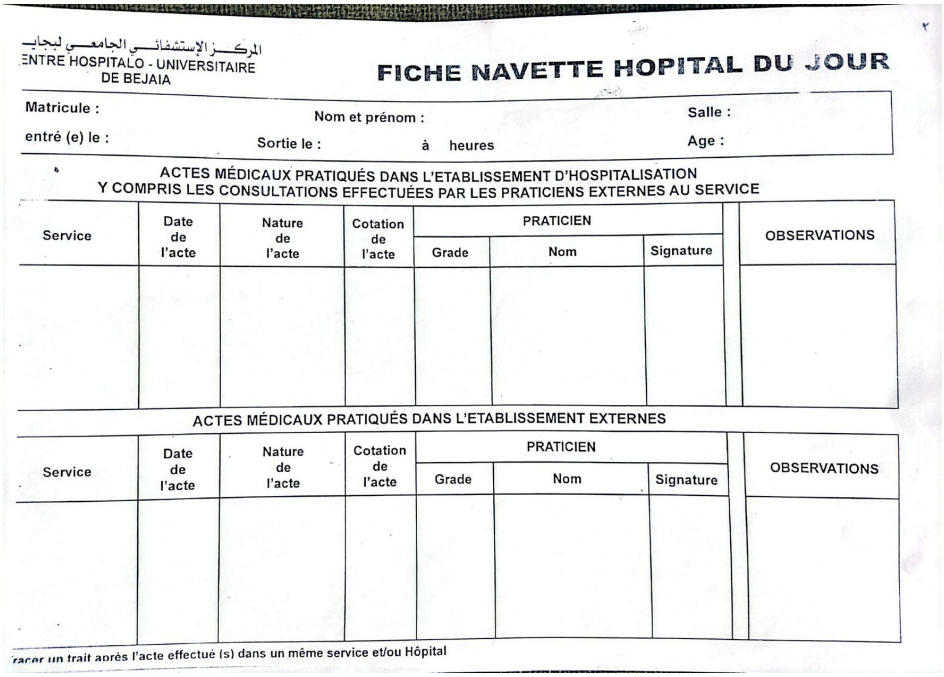
Nom et Prénom Garde du Malade : .....
Type Pièce d'identité présentée : .....

Signature,

Date et Visa du Praticien,
Le .....

Figure 4.2: Figure « Demande hospitalisation »

La figure 4.6 ci-dessous représente la fiche pour un hôpital du jour (HDJ)



المركـــز الإستشفائـــي الجامعـــي لبجايـ
ENTRE HOSPITALO - UNIVERSITAIRE
DE BEJAIA

FICHE NAVETTE HOPITAL DU JOUR

Matricule : Nom et prénom : Salle :

entré (e) le : Sortie le : à heures Age :

ACTES MÉDICAUX PRATIQUÉS DANS L'ETABLISSEMENT D'HOSPITALISATION
Y COMPRIS LES CONSULTATIONS EFFECTUÉES PAR LES PRATICIENS EXTERNES AU SERVICE

Service	Date de l'acte	Nature de l'acte	Cotation de l'acte	PRATICIEN			OBSERVATIONS
				Grade	Nom	Signature	

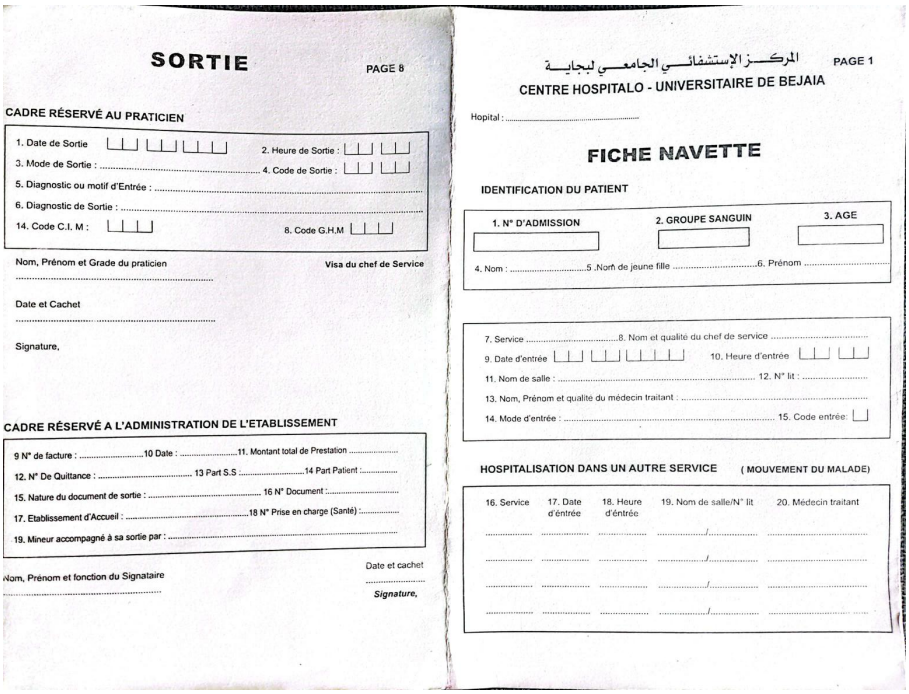
ACTES MÉDICAUX PRATIQUÉS DANS L'ETABLISSEMENT EXTERNES

Service	Date de l'acte	Nature de l'acte	Cotation de l'acte	PRATICIEN			OBSERVATIONS
				Grade	Nom	Signature	

racer un trait après l'acte effectué (s) dans un même service et/ou Hôpital

Figure 4.3: Figure « Fiche HDJ »

La figure 4.4 ci-dessous représente la fiche navette du patient pour la sortie et l'admission.



SORTIE

PAGE 8

CADRE RÉSERVÉ AU PRATICIEN

1. Date de Sortie
2. Heure de Sortie :
3. Mode de Sortie :
4. Code de Sortie :
5. Diagnostic ou motif d'Entrée :
6. Diagnostic de Sortie :
14. Code C.I. M :
8. Code G.H.M

Nom, Prénom et Grade du praticien

Visa du chef de Service

Date et Cachet

Signature,

CADRE RÉSERVÉ A L'ADMINISTRATION DE L'ETABLISSEMENT

9 N° de facture :
10 Date :
11. Montant total de Prestation
12. N° De Quittance :
13 Part S.S
14 Part Patient :
15. Nature du document de sortie :
16 N° Document :
17. Etablissement d'Accueil :
18 N° Prise en charge (Santé) :
19. Mineur accompagné à sa sortie par :

Nom, Prénom et fonction du Signataire

Date et cachet

Signature,

المركـــز الإستشفائـــي الجامعـــي لبجايـــة

PAGE 1

CENTRE HOSPITALO - UNIVERSITAIRE DE BEJAIA

Hopital :

FICHE NAVETTE

IDENTIFICATION DU PATIENT

1. N° D'ADMISSION
2. GROUPE SANGUIN
3. AGE
4. Nom :
5. Nom de jeune fille
6. Prénom

7. Service
8. Nom et qualité du chef de service
9. Date d'entrée
10. Heure d'entrée
11. Nom de salle :
12. N° lit :
13. Nom, Prénom et qualité du médecin traitant :
14. Mode d'entrée :
15. Code entrée:

HOSPITALISATION DANS UN AUTRE SERVICE (MOUVEMENT DU MALADE)

16. Service	17. Date d'entrée	18. Heure d'entrée	19. Nom de salle/N° lit	20. Médecin traitant
			/	
			/	
			/	
			/	

Figure 4.4: Figure « Fiche navette »

La figure 4.5 ci-dessous représente le document certificat de séjour.

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

WILAYA DE BEJAIA
CENTRE HOSPITALO-UNIVERSITAIRE
DE BEJAIA

CERTIFICAT DE SÉJOUR

Le directeur du centre Hospitalo-Universitaire de Béjaia
Certifie que (la) nommé (e) :.....
Agé (e) de : ..... Matricule .....Sce :.....
Demeurant à : .....
Est admis (e) le :.....

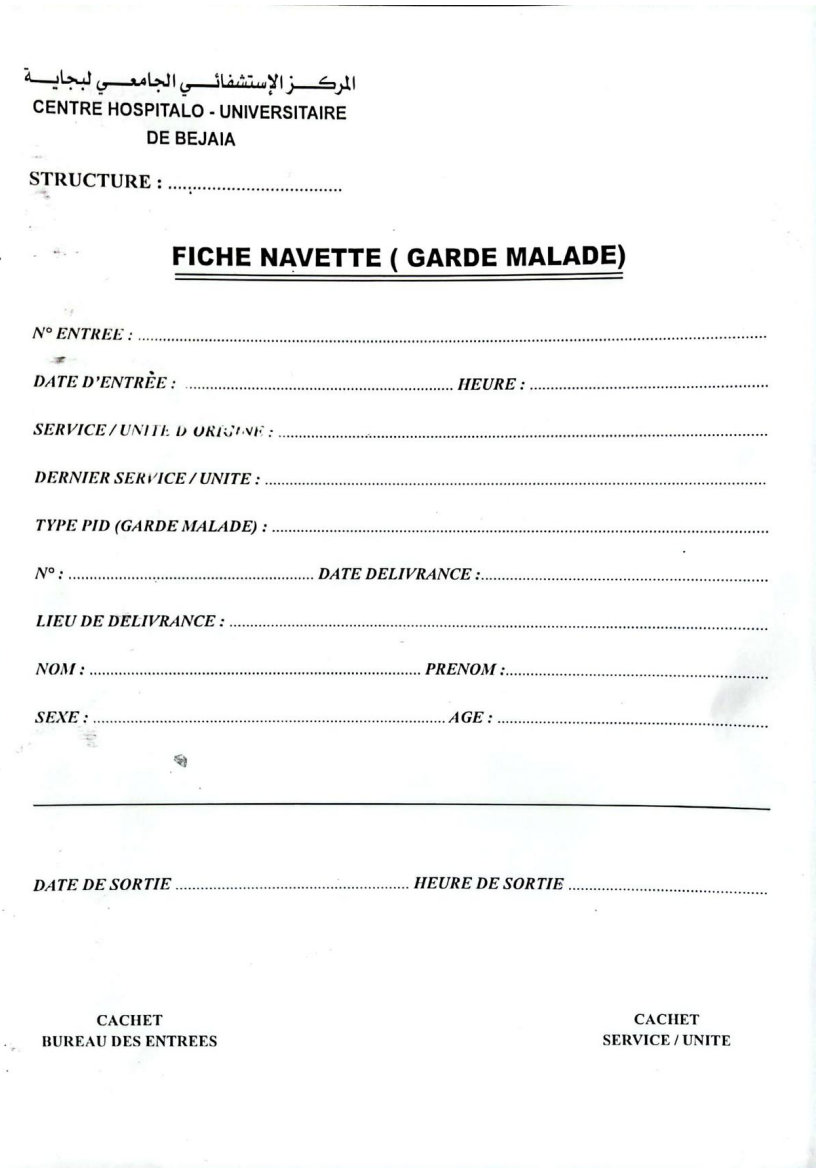
- En traitement à ce jour.
- Sortie :.....
- Transféré (e) le .....
- Décédé, (e) le.....

Béjaia, le .....

P/Le Directeur
le Préposé aux Admissions

Figure 4.5: Figure « Certificat de séjour »

La figure 4.6 ci-dessous représente la fiche navette du garde malade.



المركـــز الإستشفائـــي الجامعـــي لبجايـــة
CENTRE HOSPITALO - UNIVERSITAIRE
DE BEJAIA

STRUCTURE : .....

FICHE NAVETTE (GARDE MALADE)

N° ENTREE : .....

DATE D'ENTRÈE : ..... *HEURE :* .....

SERVICE / UNITE D ORIGINE : .....

DERNIER SERVICE / UNITE : .....

TYPE PID (GARDE MALADE) : .....

N° : ..... *DATE DELIVRANCE :*.....

LIEU DE DELIVRANCE : .....

NOM : ..... *PRENOM :*.....

SEXE : ..... *AGE :* .....

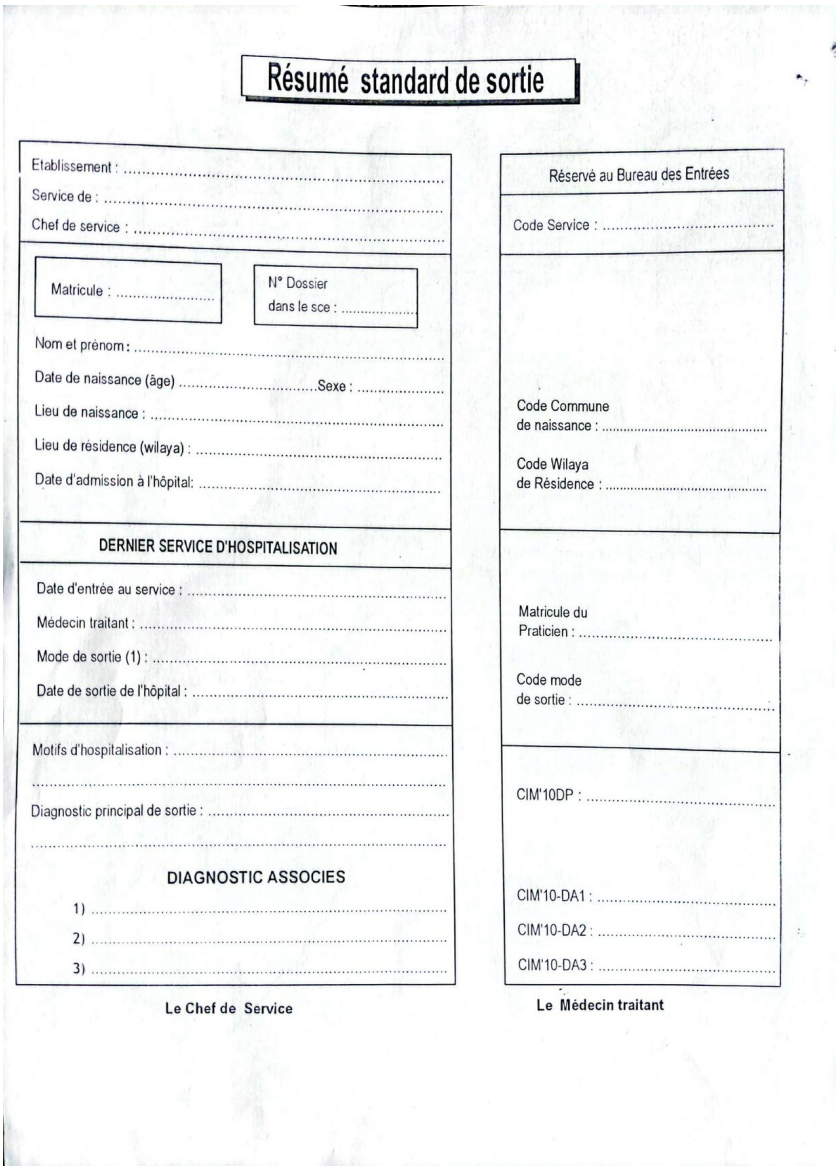
DATE DE SORTIE ..... *HEURE DE SORTIE* .....

CACHET
BUREAU DES ENTREES

CACHET
SERVICE / UNITE

Figure 4.6: Figure « Fiche navette du garde malade »

La figure 4.7 ci-dessous représente le document résumé standard de sortie.



Résumé standard de sortie

Etablissement : .....
Service de : .....
Chef de service : .....

Matricule : .....

N° Dossier dans le sce : .....

Nom et prénom : .....
Date de naissance (âge) ..... Sexe : .....
Lieu de naissance : .....
Lieu de résidence (wilaya) : .....
Date d'admission à l'hôpital: .....

DERNIER SERVICE D'HOSPITALISATION

Date d'entrée au service : .....
Médecin traitant : .....
Mode de sortie (1) : .....
Date de sortie de l'hôpital : .....

Motifs d'hospitalisation : .....
.....
Diagnostic principal de sortie : .....
.....

DIAGNOSTIC ASSOCIES

1) .....
2) .....
3) .....

Le Chef de Service

Réservé au Bureau des Entrées

Code Service : .....

Code Commune de naissance : .....

Code Wilaya de Résidence : .....

Matricule du Praticien : .....

Code mode de sortie : .....

CIM'10DP : .....

CIM'10-DA1 : .....
CIM'10-DA2 : .....
CIM'10-DA3 : .....

Le Médecin traitant

Figure 4.7: Figure « Résumé standard de sortie »

Résumés

Abstract

This end-of-study thesis, conducted as part of obtaining a Master's degree in Data Science and Decision Support, focuses on the design and implementation of a machine learning model. The main objective of this project is to predict the length of stay after exploring and cleaning the necessary data for this prediction.

To achieve these objectives, I used Python as the programming language and Jupyter as the development platform. Additionally, I chose the Random Forest algorithm for my prediction.

After completing the development, I created a new DataFrame and integrated the hospital data to explore them for future analysis.

Résumé

Ce mémoire de fin d'étude, réalisé dans le cadre de l'obtention du diplôme de Master en Science de Données et Aide à la Décision, se concentre sur la conception et la réalisation d'un modèle de machine learning. L'objectif principal de ce projet est de prédire la durée de séjour après avoir exploré et nettoyé les données nécessaires pour cette prévision.

Pour atteindre ces objectifs, on a utilisé Python comme langage et Jupyter comme plateforme de développement. De plus, on a choisi l'algorithme Random Forest pour les prévisions.

Après avoir achevé le développement, on a créé un nouveau DataFrame et intégré les données de l'hôpital afin de les explorer pour les prochaines analyses.