Université de Montréal

Faculté des arts et des sciences

Département d'informatique et de recherche opérationnelle (DIRO)

Rapport de stage

Évaluation du risque de retour à la maison

Auteur :
Vilon Saint-Fleurose
MSc, informatique
Université de Montréal

Directeur de recherche :
Dr. Michalis Famelis
Professeur adjoint
Université de Montréal

Superviseur :
Nicolas Coallier
Vice-Président Exécutif,
TIC
ML+

17 mai 2018

Université de Montréal

Table des matières

Remerciements

Je veux commencer d'abord par remercier le Grand Dieu Tout-Puissant, le Créateur de l'univers, des cieux et de la terre qui m'a donné la vie, la santé, les opportunités et tout ce dont j'avais besoin pour faire cette grande et belle étude à l'université de Montréal. Il a rendu toutes choses possibles en ma faveur, moi qui suis pécheur et désobéissant ; immérité de toutes ces grâces. Il m'accompagnait toujours dans les moments les plus difficiles de ma vie, Il ne m'a jamais laissé seul ; surtout dans les moments où je devais payer les frais de scolarité qui étaient si énormes et impossibles à payer de mon propre compte. A un Dieu si merveilleux et si bon, je Lui dois beaucoup de reconnaissance.

Je remercie aussi ma femme qui m'a beaucoup supporté pendant plus de deux années d'études. Elle n'a jamais murmuré, ni découragé quand nous devions passer par des moments difficiles de notre vie conjugale à cause de ces études. Elle a mis toutes ses ressources disponibles pour entretenir la famille et payer mes études quand j'étais moi-même dans l'impossibilité de travailler. Vraiment, ma femme est une bénédiction dans ma vie, un cadeau venant de Dieu. Je t'aime ma chérie.

Je tiens à remercier le professeur Michalis Famelis d'avoir accepté être mon directeur de recherche et supervisé ce stage, il est toujours là pour m'encourager et me pousser vers l'avant. Il répond toujours présent à tous mes appels, il est toujours disponible pour me rencontrer, me parler et me conseiller ; même en dehors du cadre universitaire.

Je remercie Nicolas Coallier et toute l'équipe ML+ qui ont accepté que je sois leur stagiaire, ils ont placé leur confiance en moi quoiqu'ils ne me connaissaient pas encore. Cette équipe, quoique jeune, est très dynamique et chaleureuse, c'est une équipe motivante qui stimule la connaissance. J'ai dû apprendre beaucoup de choses par rapport à eux. Finalement, je présente mes sincères remerciements à toute la communauté universitaire, à DIRO en particulier. Merci pour la formation prestigieuse que vous m'avez fournie. Cette formation est si solide qu'elle m'aidera rapidement à intégrer le marché du travail sans perdre de temps.

Table des figures

Liste des tableaux

Résumé

L'arthroplastie totale de la hanche et du genou réduit considérablement la douleur et améliore la fonction chez les personnes atteintes d'arthrose avancée. le vieillissement de la génération du "baby boom", combiné au désir de maintenir un mode de vie actif et sans douleur, entraînera une augmentation du nombre annuel de chirurgies de remplacement articulaire pratiquées dans États Unis [2]. L'Académie américaine des chirurgiens orthopédiques a projeté que d'ici 2030, les arthroplasties totales de la hanche et du genou augmentera à plus de 748 000 / an [13]. Avec l'augmentation continue du nombre de chirurgies de remplacement, il devient crucial aux administrateurs des hôpitaux de déterminer si les patients doivent rester à l'hôpital pour une réadaptation après leur chirurgie ou s'ils doivent être renvoyés à la maison puisque les coûts associés à une réadaptation doivent être pris en charge par les hôpitaux. Il est important de souligner que la majorité des compagnies d'assurance santé aux États Unis n'assument pas les coûts liés à la réadaptation des patients. Ces frais qui sont très élevés (entre 15-30,000 $ par patient) doivent être assurés par les hôpitaux. Pour prendre la décision de retour ou non, les chirurgiens tiennent compte des antécédents médicaux du patient, ce qui est un travail difficile à effectuer manuellement. Dans notre projet de stage, nous développerons un model de machine learning qui aide à prendre cette décision de façon automatique.

1 | Introduction générale

Le système de santé aux USA repose sur deux types de financement. Premièrement, sur le financement public qui englobe certains groupes de personnes de la population américaine uniquement. Il se concentre essentiellement sur deux programmes : le programme fédéral Medicare pour les plus de 65 ans et les personnes gravement handicapées (soit 15 % de la population) et le programme Medicaid qui s'adresse aux familles pauvres avec enfants et touche 11 % de la population. Deuxièmement, sur le financement privé qui touche tout le reste de la population, l'assurance est donc majoritairement privée aux Etats-Unis [10]. Les Américains sont assurés en général via leurs employeurs ou sinon de manière individuelle lorsque leur employeur ne propose pas d'assurance ou qu'ils travaillent en indépendant. La composante « assurance médicale » dans le choix d'un emploi est donc un critère important.

Le système américain, libéral et fondé sur le marché, s'organise autour d'assurances privées souvent liées à l'emploi et d'une assurance maladie obligatoire, liée notamment à la vieillesse et aux faibles revenus. Cependant, ce système, qui n'est pas universel, échoue à couvrir l'intégralité de la population, dont une partie se retrouve sans assurance santé. [9]

Les compagnies d'assurance privées ne couvrent pas les coûts associés à une réadaptation d'un patient après une arthroplastie (Opération ayant pour but de rétablir la forme et la mobilité d'une articulation abîmée ou bloquée.). Il revient à l'hôpital de couvrir ces coûts qui sont très élevés d'après le docteur orthopédiste Jonah Hebert-Davies qui a fait ses études de spécialité en Orthopédie à l'université de Seattle, Washington USA [12]. Selon le docteur les coûts estimés sont entre 15 à 30,000 $ par patient. Ce qui revient à une grande perte financière pour les hôpitaux. La décision de maintenir ou de renvoyer un patient après une arthroplastie devient une question préoccupante dans la mesure où elle pourrait réduire le coût budgétaire des hôpitaux. Si la décision prise est de renvoyer le patient, l'hôpital est le gagnant sinon il est le perdant.

D'autres part, l'arthrose (maladie qui touche les articulations et caractérisée par la douleur et la difficulté à effectuer des mouvements articulaires [25]) touche environ 27 millions d'adultes aux États-Unis [14]. Ce qui augmente considérablement le nombre d'arthroplastie dans les hôpitaux, et aussi le poids de travail des médecins à savoir quand est-ce qu'il ya une réadaptation ou non.

Il est un travail fastidieux pour les spécialistes orthopédistes de vérifier manuellement les données médicales d'un patient pour pouvoir décider s'il doit être renvoyé ou non. Ce travail consiste en ce que les données médicales d'un dossier patient sont très volumineuses. Créer un moyen d'automatiser ce processus est le but de notre projet.

Nous supposons qu'il existe déjà un jeu de données disponibles contenant toutes les informations touchant un grand nombre de patients et que chaque échantillon de ce jeu de données est un dossier patient. Nous supposons aussi que ce jeu de données possède des caractéristiques suffisantes pour décider si un patient doit être renvoyé ou non. Nous considérons aussi que ce jeu de données est composé de patients qui font l'objet d'une décision de retour. Si toutes ces conditions sont réunies, nous pouvons profiter des différentes technologies de la science de données en général et du marching learning en particulier pour développer un outil d'aide à la décision qui aide les orthopédistes à prendre la décision de retour automatiquement.

La suite de ce rapport est composé de plusieurs chapitres et chaque chapitre a des sections, subsections etc. Le chapitre 2 donne un contexte théorique des différents éléments qui sont importants pour notre projet. Le chapitre 3 présente en détail la description technique des différentes étapes de notre projet. Le chapitre 4 fait une évaluation de notre outil crée. Le chapitre 5 présente les limitations et les travaux futurs. Le chapitre 6 détaille les travaux connexes et le chapitre 7 est celui de la conclusion.

2 | Contexte théorique

2.1 Introduction

Dans ce chapitre nous voulons présenter une approche théorique des différents concepts que nous avons utilisés dans le cadre notre projet. Nous allons faire un résumé de chaque concept important que nous avons utilisé dans le cadre de l'implémentation de ce projet. Nous n'allons pas toucher tout ce qui a rapport aux concepts mais juste ce qui nous intéresse et util pour le projet. Il y aura des références pour ceux qui veulent approfondir les concepts.

2.2 Sciences des données

La science des données est un domaine interdisciplinaire de méthodes, processus, algorithmes et systèmes scientifiques permettant d'extraire des connaissances ou des informations à partir de données sous diverses formes, structurées ou non [26]. Elle emploie des techniques et des théories tirées de plusieurs autres domaines plus larges des mathématiques, la statistique principalement, la théorie de l'information et la technologie de l'information, notamment le traitement de signal, des modèles probabilistes, l'apprentissage automatique, l'apprentissage statistique, la programmation informatique, l'ingénierie de données, la reconnaissance de formes et l'apprentissage, la visualisation, l'analytique prophétique, la modélisation d'incertitude, le stockage de données, la compression de données et le calcul à haute performance. Les méthodes qui s'adaptent aux données de masse sont particulièrement intéressantes dans la science des données, bien que la discipline ne soit généralement pas considérée comme limitée à ces données.

L'objectif du « data scientist » (expert en données massives) est de produire des méthodes (automatisées, autant que possible) de tri et d'analyse de données de masse et de sources plus ou moins complexes ou disjointes de données, afin d'en extraire des informations utiles ou potentiellement utiles. Le métier de data scientist est apparu pour trois raisons principales [15] :

- l'explosion de la quantité de données produites et collectées par les humains ;
- l'amélioration et l'accessibilité plus grande des algorithmes de machine learning ;
- l'augmentation exponentielle des capacités de calcul des ordinateurs.

Le cycle de travail du data scientist comprend notamment :

- la récupération des données utiles à l'étude ;
- le nettoyage des données pour les rendre exploitables ;
- une longue phase d'exploration des données afin de comprendre en profondeur l'articulation des données ;
- la modélisation des données ;
- l'évaluation et interprétation des résultats ;
- la conclusion de l'étude : prise de décision ou déploiement en production du modèle.

La figure 2.1 nous donne une vue générale du travail d'un data scientist. Étant stagiaire en science de données, nous avons suivi minutieusement, dans le cadre de notre projet, le cycle de travail complet du data scientist. Depuis la récupération des données jusqu'au déploiement d'un système en production. Nous expliquerons en détail dans la section "Aperçu", comment nous implémentons ce cycle au sein de notre projet.

Deux composantes sont nécessaires pour pouvoir commencer à se demander si la data science peut, oui ou non, apporter de la valeur et aider à la résolution d'un problème : des **données** et une **problématique** bien définie [15].

2.2.1 Données.

Les données constituent la ressource principale pour qu'un data scientist puisse effectuer son travail correctement. Nos données proviennent du département orthopédique de l'université de Seattle, Washington, États-Unis. Elles sont au nombre de 2718 files et 92 colonnes. Nous les utilisons comme données historiques (*historical data, en anglais*) dans le cadre du développement de notre modèle de machine learning.

2.2.2 Problématique.

La problématique de notre projet, comme nous l'avons vu dans l'introduction, est d'aider à décider si un patient doit rester à l'hôpital ou rentrer chez lui après une arthroplastie.

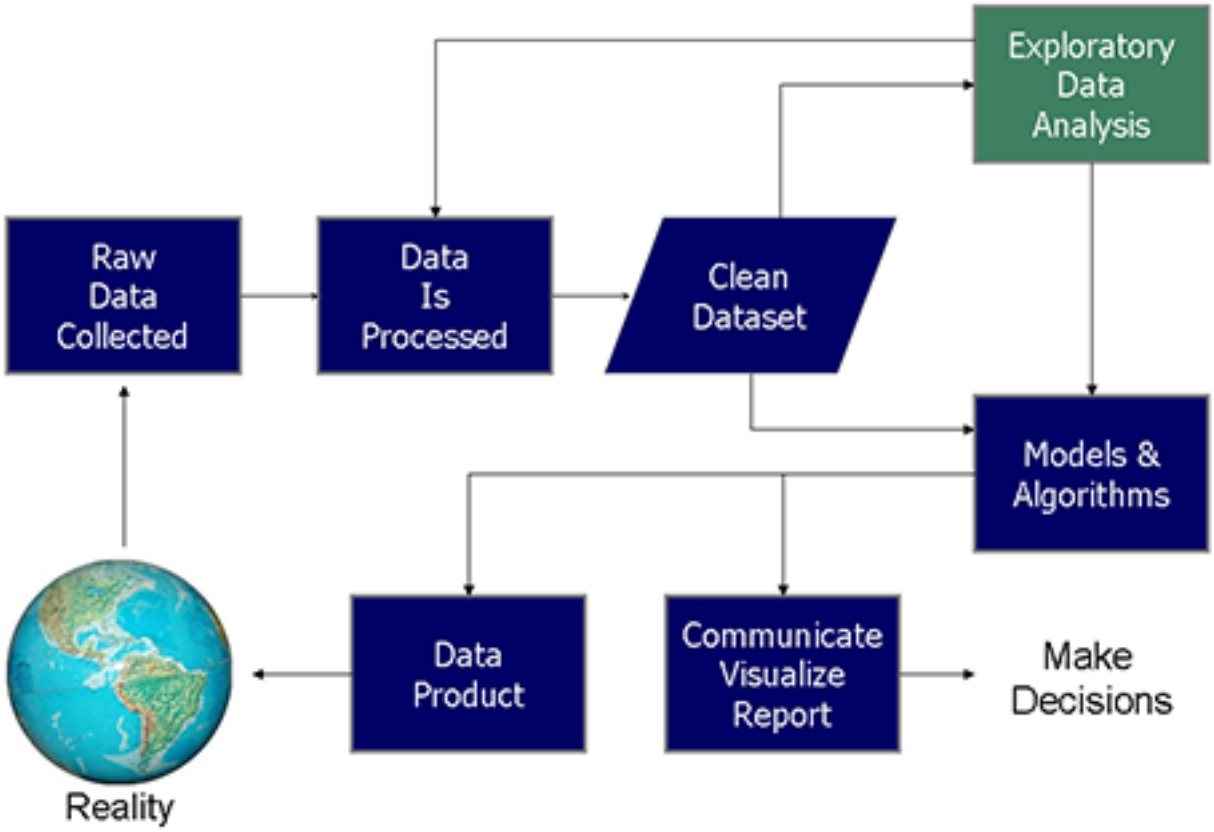

Fig. 2.1 – Work cycle of data scientist [15]

Au sein de ce cycle, le machine learning désigne l'ensemble des méthodes de modélisation statistique à partir des données, et se situe bien au coeur du travail du data scientist. Dans la sous-section qui suit nous allons aborder le machine learning, car le gros de notre travail est de construire un model machine learning d'aide à la décision. Nous verrons une brève introduction sur ce quoi le machine learning, ensuite un survol sur les concepts fondamentaux du ML et finalement nous nous concentrerons sur un sous-ensemble de concepts qui touchent notre projet, principalement le problème de classification. Nous n'aborderons pas dans ce rapport, la théorie sur l'exploration des données *(data mining en anglais)*. Nous supposons que ceux qui lisent ce rapport ont déjà des notions élémentaires sur cette question, sinon ils peuvent visiter wikipedia ou n'importe quelle autre page

pour avoir des idées générales sur ce point.

2.3 Vue d'ensemble sur Machine Learning

Dans la section précédente, nous avons pu y voir plus clair sur le cycle global de travail du data scientist. Nous allons maintenant parler du machine learning dans cette section, c'est à dire la modélisation des données. Nous utilisons le machine learning probablement des dizaines de fois par jour sans même le savoir. Chaque fois que nous effectuons une recherche Web sur Google ou Bing, cela fonctionne si bien c'est parce que leur logiciel de machine learning a trouvé comment classer les pages. Lorsque Facebook ou l'application photo d'Apple reconnaît nos amis dans nos images, c'est aussi du machine learning. Chaque fois que nous lisons notre courrier électronique et qu'un filtre anti-spam nous évite d'avoir à parcourir des tonnes de spam, c'est parce que nos ordinateurs ont appris à distinguer le spam du courrier non-spam. Donc, c'est du machine learning. C'est la science qui consiste à apprendre les ordinateurs sans être explicitement programmés. La figure 2.2 nous montre en grosso modo les techniques du machine learning, à un très haut d'abstraction.

Le machine learning constitue une manière de modéliser des phénomènes, dans le but de prendre des décisions stratégiques. Les algorithmes utilisés permettent, dans une certaine mesure, à un système piloté par ordinateur (un robot éventuellement), ou assisté par ordinateur, d'adapter ses analyses et ses comportements en réponse, en se fondant sur l'analyse de données empiriques provenant d'une base de données ou de capteurs. La difficulté réside dans le fait que l'ensemble de tous les comportements possibles compte tenu de toutes les entrées possibles devient rapidement trop complexe à décrire **(on parle d'explosion combinatoire)**. On confie donc à des programmes le soin d'ajuster un modèle pour simplifier cette complexité et de l'utiliser de manière opérationnelle. Idéalement, l'apprentissage visera à être non supervisé, c'est-à-dire que la nature des données d'entrainement n'est pas connue [23].

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient :

2.3.1 Apprentissage supervisé

L'apprentissage supervisé (supervised learning en anglais) est une technique d'apprentissage automatique où l'on cherche à produire automatiquement des règles à partir d'une base de données d'apprentissage contenant des « exemples » (en général des cas déjà traités et validés).

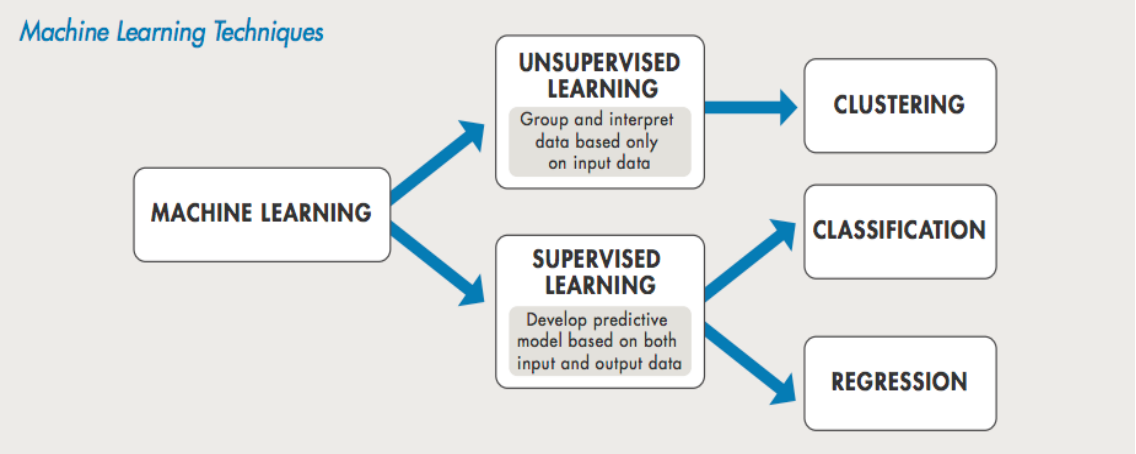

Fig. 2.2 – Machine learning representation [21]

Une base de données d'apprentissage (ou ensemble d'apprentissage) est un ensemble de couples entrée-sortie $(x_n, y_n)_{1 \leq n \leq N}$ avec $x_n \in X$ et $y_n \in Y$, que l'on considère être tirées selon une loi sur $X \times Y$ fixe et inconnue, par exemple x_n suit une loi uniforme et $y_n = f(x_n) + w_n$ où w_n est un bruit centré.

On distingue trois types de problèmes solubles avec une méthode d'apprentissage automatique supervisée [24] :

- $Y \subset \mathbb{R}$: lorsque la sortie que l'on cherche à estimer est une valeur dans un ensemble continu de réels, on parle d'un problème de régression. La fonction de prédiction est alors appelée un régresseur.
- $Y = \{1, \cdots, I\}$ lorsque l'ensemble des valeurs de sortie est fini, on parle d'un problème de classification, qui revient à attribuer une étiquette à chaque entrée. La fonction de prédiction est alors appelée un classificateur (ou classificateur).
- Lorsque Y est un ensemble de données structurées, on parle d'un problème de prédiction structurée, qui revient à attribuer une sortie complexe à chaque entrée. Par exemple, en bio-informatique le problème de prédiction de réseaux d'interactions entre gènes peut être considéré comme un problème de prédiction structurée dans laquelle l'ensemble possible des sorties structurées est l'ensemble de tous les graphes modélisant les interactions possibles [24].

2.3.2 Apprentissage non supervisé

Quand le système ou l'opérateur ne disposent que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés,

on parle d'apprentissage non supervisé ou clustering. Aucun expert n'est requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le partitionnement de données, data clustering en anglais, est un algorithme d'apprentissage non supervisé.
Le système doit ici — dans l'espace de description (la somme des données) — cibler les données selon leurs attributs disponibles, pour les classer en groupe homogènes d'exemples. La similarité est généralement calculée selon une fonction de distance entre paires d'exemples. C'est ensuite à l'opérateur d'associer ou déduire du sens pour chaque groupe et pour les motifs (patterns en anglais) d'apparition de groupes, ou de groupes de groupes, dans leur « espace ». Divers outils mathématiques et logiciels peuvent l'aider. On parle aussi d'analyse des données en régression (ajustement d'un modèle par une procédure de type moindres carrés ou autre optimisation d'une fonction de coût). Si l'approche est probabiliste (c'est-à-dire que chaque exemple, au lieu d'être classé dans une seule classe, est caractérisé par un jeu de probabilités d'appartenance à chacune des classes), on parle alors de « soft clustering » (par opposition au « hard clustering »).
Cette méthode est souvent source de sérendipité(le fait de réaliser une découverte scientifique ou une invention technique de façon inattendue à la suite d'un concours de circonstances fortuit)
ex. : Pour un épidémiologiste qui voudrait dans un ensemble assez large de victimes de cancer du foie tenter de faire émerger des hypothèses explicatives, l'ordinateur pourrait différencier différents groupes, que l'épidémiologiste chercherait ensuite à associer à divers facteurs explicatifs, origines géographique, génétique, habitudes ou pratiques de consommation, expositions à divers agents potentiellement ou effectivement toxiques (métaux lourds, toxines telle que l'aflatoxine, etc.) [23].

2.3.3 Apprentissage semi-supervisé

Effectué de manière probabiliste ou non, il vise à faire apparaître la distribution sous-jacente des exemples dans leur espace de description. Il est mis en œuvre quand des données (ou « étiquettes ») manquent... Le modèle doit utiliser des exemples non étiquetés pouvant néanmoins renseigner. Ex. : En médecine, il peut constituer une aide au diagnostic ou au choix des moyens les moins onéreux de tests de diagnostic. Apprentissage partiellement supervisé probabiliste ou non, quand l'étiquetage des données est partiel3. C'est le cas quand un modèle énonce qu'une donnée n'appartient pas à une classe A, mais peut-être à une classe B ou C (A, B et C étant 3 maladies par exemple évoquées dans le cadre d'un diagnostic différentiel) [23].

2.3.4 Apprentissage par renforcement

l'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage [23]. ex. : L'algorithme de Q-learning est un exemple classique.

2.3.5 Apprentissage par transfert

L'apprentissage par transfert peut être vu comme la capacité d'un système à reconnaître et appliquer des connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes. La question qui se pose est : comment identifier les similitudes entre la ou les tâche(s) cible(s) et la ou les tâche(s) source(s), puis comment transférer la connaissance de la ou des tâche(s) source(s) vers la ou les tâche(s) cible(s) ? [23]

2.3.6 Notion de dataset

Le dataset est l'ensemble des données utilisé pour entraîner un modèle. Il existe deux types de jeux de données (dataset) généraux. Jeux étiqueté et jeux sans étiquette [4].

2.3.7 Training Set and Test Set

Dans l'apprentissage automatique, un jeu de données universel inconnu est supposé exister, qui contient toutes les paires de données possibles ainsi que leur distribution de probabilité d'apparition dans le monde réel. Alors que dans les applications réelles, ce que nous avons observé est seulement un sous-ensemble de l'ensemble de données universel en raison du manque de mémoire ou d'un autre inévitable les raisons. Cet ensemble de données acquises s'appelle l'ensemble d'apprentissage **(Training Set)** et est utilisé pour apprendre les propriétés et la connaissance de l'ensemble de données universel. Afin d'examiner la performance de l'apprentissage, un autre ensemble de données peut être réservé pour le test, appelé ensemble de test ou données de test **(Test Set)** [4].

Le machine learning est un champ d'études vaste, c'est un domaine de recherche. Dans ce rapport nous ne pouvons pas aborder tous les concepts découlant du machine learning mais nous allons considérer un sous-ensemble de concepts que nous avons utilisés pour le développement de notre projet. Dans notre cas, c'est un problème de classification qui fait partie de l'apprentissage supervisé. C'est comme l'exemple des courriers spam et anti-spam que nous avons pris dans l'introduction

de cette section. Le problème est de décider est-ce que le patient doit retourner chez lui ou s'il doit rester à l'hôpital après une arthoplastie. Si la réponse est vraie le patient reste à l'hôpital sinon il retourne chez lui. Il faut toujours garder en esprit que cette décision est importante pour un centre orthopédique car elle peut réduire ou augmenter le coût budgétaire.

2.4 Problème de Classification

Dans l'apprentissage automatique et les statistiques, le problème de classification est le problème d'identifier lequel d'un ensemble de catégories (sous-population) appartient à une nouvelle observation, sur la base d'un ensemble de données contenant des observations (ou instances) dont la composition est connue [27].

Voici quelques exemples de problèmes de classification. Nous avons déjà parlé de la classification du spam par courrier électronique comme exemple de problème de classification. Un autre exemple serait le classement des transactions en ligne. Donc, si vous avez un site Web qui vend des trucs et si vous voulez savoir si une transaction particulière est frauduleuse ou non, si quelqu'un utilise une carte de crédit volée ou a volé le mot de passe de l'utilisateur. Si la transaction est frauduleuse, on retourne 1 comme valeur sinon l'algorithme retourne 0.
0 est aussi appelé la classe négative, et 1 la classe positive, et ils sont parfois aussi désignés par les symboles "-" et "+".

Dans notre projet, le problème est un problème de classification binaire parce qu'on doit décider si oui ou non le patient reste à l'hôpital après une intervention chirurgicale. Si le patient reste, la valeur de la classe est positive (c'est-à-dire 1), s'il ne reste pas ou il est renvoyé, la valeur de la classe est négative (c'est-à-dire 0).

2.4.1 Mesure de performance des algorithmes de classification

L'évaluation de la performance des méthodes d'apprentissage automatique est aussi cruciale que l'algorithme lui-même, car il identifie les forces et les faiblesses de chaque algorithme d'apprentissage. Différentes mesures de performance sont utilisées pour évaluer différents algorithmes d'apprentissage automatique. Pour l'instant, nous allons nous concentrer sur ceux utilisés pour les problèmes de classification. Quelques metrics que nous pouvons utiliser pour évaluer la performance des problèmes de classification sont Log-Loss, Accuracy, AUC(Area under Curve) etc. Ils font partie des plus communs mais nous pouvons créer des metrics personnalisés en fonction de nos besoins [19].

Les metrics que vous choisissez pour évaluer votre modèle d'apprentissage automatique sont très importantes. Le choix des paramètres influence la façon dont la performance des algorithmes d'apprentissage automatique est mesurée et comparée. Pour ne pas perdre plus de temps, voyons ce que sont quelques metrics que nous avons utilisé dans notre projet.

2.4.2 Matrice de confusion

La matrice de confusion **(Confusion Matrix en anglais)** est l'une des mesures les plus intuitives et les plus faciles (à moins bien sûr, vous n'êtes pas confus) utilisées pour trouver l'exactitude et la précision du modèle. Il est utilisé pour les problèmes de classification où la sortie peut être de deux types ou plus de classes [19].

La figure 2.3 est une représentation de la matrice de confusion. La valeur actuelle (Actual value) représente la vraie valeur de l'étiquette tandis que la valeur prédite (Predicted value) représente la valeur retournée par le modèle machine learning.

True Positives (TP) : Les vrais positifs sont les cas où la classe réelle du point de données était 1 (Vrai) et la prédiction est également 1 (Vrai)

True Negatives (TN) : Les vrais négatifs sont les cas où la classe réelle du point de données était 0 (Faux) et la prédiction est également 0 (Faux)

False Positives (FP) : Les faux positifs sont les cas où la classe réelle du point de données était 0 (Faux) et la prédiction est 1 (Vrai). Faux parce que le modèle a prédit incorrectement et positivement parce que la classe prédite était positive. (1)

False Negatives (FN) : Les faux négatifs sont les cas où la classe réelle du point de données était 1 (Vrai) et la prédiction est 0 (Faux). Faux parce que le modèle a prédit incorrectement et négativement parce que la classe prédite était négative. (0)

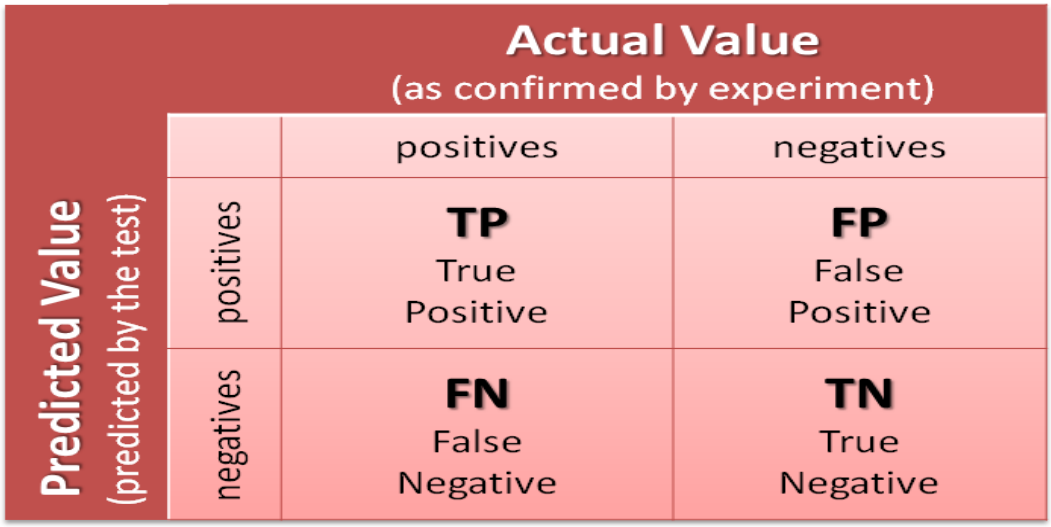

Fig. 2.3 – Confusion Matrix representation [20]

2.4.3 Calcul de certains metrics de classification à partir de la matrice confusion

La matrice de confusion en soi n'est pas une mesure de performance en tant que telle, mais presque toutes les métriques de performance sont basées sur la matrice de confusion et les nombres qui s'y trouvent. Nous allons voir comment utiliser la matrice de confusion pour calculer quelques metrics de classification [19].

1. **Accuracy :** L'exactitude **(Accuracy en anglais)** dans les problèmes de classification est le nombre de prédictions correctes faites par le modèle sur toutes les prédictions faites. Accuracy = $\frac{TP+TN}{TP+FP+FN+TN}$
2. **Precision :** La précision permet de répondre à la question suivante : Quelle proportion d'identifications positives était effectivement correcte ? Precision = $\frac{TP}{TP+FP}$
3. **Recall or Sensitivity :** Le rappel permet de répondre à la question suivante : Quelle proportion de résultats positifs réels a été identifiée correctement ? Recall = $\frac{TP}{TP+FN}$
4. **F1-score :** Nous ne voulons pas vraiment avoir à la fois la précision et le rappel dans nos poches chaque fois que nous faisons un modèle pour résoudre un problème de classification. Donc, il est préférable que nous puissions obtenir un seul score qui représente à la fois la précision (P) et le rappel (R)
   F1score = $\frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}} = 2 \times \frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} = \frac{TP}{TP+\frac{FN+FP}{2}}$
5. **ROC-AUC :** Une courbe ROC (receiver operating characteristic) est un

graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs **(True Positifs en anglais)** en fonction du taux de faux positifs **(False Positifs en anglais)** [6].

2.5 Classes déséquilibrées

Nous terminerons le chapitre sur la classification en abordant un sujet très important en machine learning, surtout dans le problème de la classification qui est "*Jeux de données déséquilibré*" **(Imbalanced datasets or Imbalanced Classification or Imbalanced Classes, ses appellations en anglais)**. C'est un problème très commun en machine learning, spécialement en classification. Nous abordons ce sujet parce que c'était l'une des principales difficultés de notre projet.

Une distribution de classe déséquilibrée est un scénario où le nombre d'observations appartenant à une classe est significativement inférieur à celui des autres classes. Ce problème prédomine dans les cas où la détection d'anomalies est cruciale comme le vol d'électricité, les transactions frauduleuses dans les banques, l'identification de maladies rares, etc. Dans ce cas, le modèle prédictif développé avec des algorithmes conventionnels pourrait être biaisé et inexact.
Cela arrive parce que les algorithmes d'apprentissage automatique sont généralement conçus pour améliorer la précision en réduisant l'erreur. Ainsi, ils ne tiennent pas compte de la répartition / proportion des classes ou de l'équilibre des classes [22].

Plusieurs techniques ont été proposées pour manipuler les classes déséquilibrées. Voyons un bref résumé de quelques-unes d'entre elles [22] :

2.5.1 Random Under-Sampling :

Le Random Under-Sampling vise à équilibrer la distribution des classes en éliminant de manière aléatoire les exemples de classes majoritaires. Ceci est fait jusqu'à ce que les instances de la majorité et de la classe minoritaire soient équilibrées.

2.5.2 Random Over-Sampling :

Le Random Over-Sampling augmente le nombre d'instances dans la classe minoritaire en les reproduisant aléatoirement afin de présenter une représentation plus élevée de la classe minoritaire dans l'échantillon.

2.5.3 Cluster-Based Over Sampling :

Dans ce cas, l'algorithme de clustering K-means est appliqué indépendamment aux instances de classe minoritaire et majoritaire. Cela permet d'identifier les clusters dans l'ensemble de données. Par la suite, chaque grappe est suréchantillonnée de sorte que tous les clusters de la même classe ont un nombre égal d'instances et toutes les classes ont la même taille.

2.5.4 Synthetic Minority Over-sampling Technique(SMOTE) :

Cette technique est suivie pour éviter le surapprentissage qui se produit lorsque des répliques exactes d'instances minoritaires sont ajoutées à l'ensemble de données principal. Un sous-ensemble de données est pris à partir de la classe minoritaire à titre d'exemple, puis de nouvelles instances similaires synthétiques sont créées. Ces instances synthétiques sont ensuite ajoutées à l'ensemble de données d'origine. Le nouvel ensemble de données est utilisé comme un échantillon pour former les modèles de classification.

2.5.5 Modified synthetic minority oversampling technique (MSMOTE) :

C'est une version modifiée de SMOTE. SMOTE ne tient pas compte de la distribution sous-jacente de la classe minoritaire et des bruits latents dans l'ensemble de données. Pour améliorer les performances de SMOTE, une méthode modifiée MSMOTE est utilisée.

Dans notre projet nous avons opté pour la technique de SMOTE parce qu'elle permet d'atténuer le problème de sur-adaptation **(overfitting en anglais)** causé par le sur-échantillonnage aléatoire **(random oversampling, en anglais)**

2.6 Conclusion

Le chapitre 2 a présenté une vue générale sur les différents concepts utilisés en sciences de données et en machine learning. On a donné une brève définition de chaque concept tout en essayant d'être le plus clair et précis possible. Nous avons vu un sous-ensemble de ces concepts car ces champs d'études sont très vastes et contiennent beaucoup de choses intéressantes, mais nous avons mis accent sur ce qui nous intéresse pour notre projet. Par exemple notre projet est un problème de classification en machine learning, pour cela nous avons dédié une section à l'étude

brève de classification. Aussi notre dataset est déséquilibré, par conséquent nous avons eu une section qui montre comment manipuler les classes déséquilibrées et les différentes techniques de manipulation. Le chapitre 3 qui suit présentera les détails techniques de notre projet, le but est de faire sortir les différentes étapes que nous avons suivies pour arriver à un produit fini et utilisable.

3 | Description technique de notre projet

3.1 Introduction

Pour développer notre projet, nous avons fait le cycle complet d'un data scientist. Depuis l'obtention des données jusqu'au déploiement d'un système en production. Dans ce chapitre nous allons parcourir toutes les étapes du développement de notre projet.

3.2 Obtention des données

Une fois qu'on décide à attaquer un problème de machine learning, la première chose à faire est d'explorer toutes les pistes possibles pour récupérer les données. En effet, les données constituent l'expérience, les exemples qu'on va fournir à notre algorithme de machine learning afin qu'il puisse apprendre et devenir plus performant.
Les données que nous utilisons dans ce projet proviennent de la compagnie pour laquelle nous faisons le stage. Cette dernière les avait obtenues de l'université Seattle, Washington, USA. Les données sont disponibles dans un fichier d'extension csv et hébergées sur le serveur de la compagnie. Par conséquent, nous avons un accès direct aux données sans avoir besoin de les télécharger à partir d'un dépôt de Machine Learning.

3.3 Exploration des données

Le but de l'analyse exploratoire est de «connaître» l'ensemble de données. Si vous le faites à l'avance, le reste du projet sera beaucoup plus fluide, de trois façons principales [17] :

1. Vous obtiendrez des conseils précieux pour le nettoyage des données **(Data Cleaning en anglais)** qui peuvent faire ou défaire vos modèles.
2. Vous penserez à des idées pour l'ingénierie des fonctionnalités **(Feature Engineering en anglais)** qui peuvent prendre vos modèles de bonne à grande.
3. Vous obtiendrez une «idée» de l'ensemble de données, ce qui vous aidera à communiquer les résultats et à avoir un impact plus important.

Pour l'analyse exploratoire de nos données, Tout d'abord, nous commençons par une analyse basique de l'ensemble de nos données **(dataset in english)**. La table 3.1 donne une description de notre dataset utilisé pour notre projet et la figure 3.1 donne une vue de notre ensemble de données, leur représentation, et leurs valeurs possibles. Voyons une brève description de chaque composante du dataset après une analyse basique de notre exploration :

Number of Features : Représente le nombre de caractéristiques ou colonnes de notre dataset. Les features sont importants pour un modèle de machine learning puisque le modèle les utilise pour prédire des événements.

Number of Features	92
Number of Observations	2718
Number of Numerical Features	87
Number of Categorical Features	5
Target value	Discharge

TABLE 3.1 – Gives a description of our dataset

	AGE_AT_ADMIT	ASA_SCORE	Female	Height	Weight	BMI	PreOpHgb	PreOpCr	PreOpGlucose	Warfarin	Discharge
0	57.073238	1.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1
1	87.006160	2.0	1.0	154.9	60.7	25.297981	NaN	NaN	NaN	0.0	1
2	77.623546	3.0	0.0	177.8	90.7	28.690874	NaN	NaN	NaN	0.0	1
3	69.054073	2.0	1.0	162.5	68.9	26.092308	NaN	NaN	NaN	0.0	1
4	53.396304	2.0	1.0	167.6	83.9	29.868536	NaN	NaN	NaN	0.0	1

Fig. 3.1 – An overview of raw data from our dataset

Number of Observations : Représente le nombre d'échantillons **(samples or rows en anglais)** de notre ensemble de données.

Numerical Features : Ce sont les caractéristiques numériques de notre dataset. Les valeurs continues de notre dataset.

Categorical Features : Ce sont les features qui viennent sous forme de texte. Plus loin dans la section de la transformation, nous convertirons ces features en données numériques car la plupart des algorithmes de machine learning ne reconnaissent pas les données discrètes ou textuelles.

Target value : C'est l'étiquette (**label or target en anglais)** de notre ensemble de données. L'étiquette est une valeur très importante pour les algorithmes de classification car elle permet d'évaluer la valeur booléenne d'une décision. Dans notre cas c'est le *Discharge* qui permet de dire si oui ou non le patient doit rester à l'hôpital pour une réadaptation.

La table 3.2 donne une description de quelques features de notre dataset. Le nombre total des features est 92 mais nous considérons un échantillon de 12 features qui sont les plus corrélés au target. La colonne *datatype* représente le type de données et les colonnes. Et les colonnes non-null et null représentent le nombre d'échantillons non-nulls et le nombre d'échantillons nulls.

Feature name	**datatype**	**number of non-null**	**number of null**
Side	Categorical	2718	0
RawDx	Categorical	2718	0
AGE_AT_ADMIT	Numercial	2718	0
ASA_SCORE	Numercial	2716	2
Female	Boolean	2718	0
Height	Numercial	2293	425
Weight	Numercial	2301	417
BMI	Numercial	2283	435
PreOpHgb	Numercial	272	2446
PreOpCr	Numercial	255	2463
PreOpGlucose	Numercial	253	2465
Warfarin	Numercial	2482	236

TABLE 3.2 – Some features of our dataset

La table 3.3, quant à elle, c'est la représentation de notre variable target value. Dans le dataset, elle s'appelle *Discharge*. On peut remarquer que les classes positives sont significativement plus nombreuses que les classes négatives, ce qui produit un déséquilibre au niveau du dataset. Nous avons déjà discuté dans la section *"Classes déséquilibrées"*, les différentes techniques pour équilibrer les classes déséquilibrées.

Target's name	Discharge
Positive classes	2421
Negative classes	297
Total classes	2718

TABLE 3.3 – Important details about target value

3.3.1 Correlation entre les données

Nous terminons la section d'exploration de données avec les corrélations. Les corrélations nous permettent d'examiner les relations entre les entités numériques et d'autres entités numériques **(Numerical features en anglais)**. La corrélation est une valeur comprise entre -1 et 1 qui représente à quel point deux entités se déplacent à l'unisson [17].

Une corrélation positive signifie que lorsqu'une caractéristique augmente, l'autre augmente. Par exemple. Le BMI d'un patient et son poids **(Weight dans le dataset)**.

La corrélation négative signifie que lorsqu'une caractéristique augmente, l'autre diminue. Par exemple. heures consacrées à l'étude et nombre de participants. Les corrélations proches de -1 ou 1 indiquent une relation forte. Les plus proches de 0 indiquent une relation faible. 0 indique aucune relation.

Table of correlation between some features of our dataset

	AGE_AT_ADMIT	ASA_SCORE	Female	Height	Weight	BMI	PreOpHgb	PreOpCr	PreOpGlucose	Warfarin	Discharge
AGE_AT_ADMIT	1	0.081	0.12	-0.22	-0.22	-0.11	-0.2	0.17	0.16	0.044	-0.25
ASA_SCORE	0.081	1	0.027	-0.035	0.037	0.064	-0.21	0.04	0.058	0.03	-0.057
Female	0.12	0.027	1	-0.7	-0.35	0.049	-0.31	-0.24	0.073	-0.029	-0.16
Height	-0.22	-0.035	-0.7	1	0.48	-0.11	0.26	0.24	-0.0078	0.029	0.15
Weight	-0.22	0.037	-0.35	0.48	1	0.81	0.24	0.09	-0.047	0.022	0.063
BMI	-0.11	0.064	0.049	-0.11	0.81	1	0.087	-0.053	-0.012	0.021	-0.023
PreOpHgb	-0.2	-0.21	-0.31	0.26	0.24	0.087	1	-0.2	-0.14	-0.18	0.2
PreOpCr	0.17	0.04	-0.24	0.24	0.09	-0.053	-0.2	1	-0.036	0.06	-0.14
PreOpGlucose	0.16	0.058	0.073	-0.0078	-0.047	-0.012	-0.14	-0.036	1	0.04	-0.24
Warfarin	0.044	0.03	-0.029	0.029	0.022	0.021	-0.18	0.06	0.04	1	-0.11
Discharge	-0.25	-0.057	-0.16	0.15	0.063	-0.023	0.2	-0.14	-0.24	-0.11	1

Fig. 3.2 – General view of the correlation between some features of our dataset

La figure 3.2 nous donne une idée comment nos variables sont corrélées entre elles. Par exemple, le poids **(Weight dans le dataset)** et le BMI ont une correlation de 0.81 très proche de 1. On peut déduire que les patients qui ont leur BMI plus élevé sont les plus pesants. L'objectif de la correlation est de gagner en intuition sur les données, ce qui nous aidera tout au long du flux de travail.

3.4 Nettoyage des données

Le nettoyage des données **(Data cleaning en anglais)** est l'une de ces choses que tout le monde fait mais dont personne ne parle vraiment. Bien sûr, ce n'est pas la partie la plus intéressante de l'apprentissage automatique. Et non, il n'y a pas de trucs cachés et de secrets à découvrir.

Cependant, un nettoyage correct des données peut faire ou défaire notre projet. Les spécialistes des données professionnelles consacrent généralement une très grande partie de leur temps à cette étape. En fait, si vous avez un jeu de données correctement nettoyé, même des algorithmes simples peuvent apprendre des informations impressionnantes à partir des données ! [16].

3.4.1 Observations indésirables

La première étape du nettoyage des données consiste à supprimer les observations indésirables de votre jeu de données. Cela inclut des observations en double **(Duplicate observations)** ou non pertinentes **(Irrelevant observations)** [16].

3.4.2 Duplicate observations :

Les observations en double surviennent le plus souvent lors de la collecte de données, notamment lorsque vous combinez des ensembles de données provenant de plusieurs endroits, capturez les données et recevez des données de clients / autres départements [16].

3.4.3 Irrelevant observations :

Les observations non pertinentes sont celles qui ne correspondent pas vraiment au problème spécifique que vous essayez de résoudre [16].

Dans notre projet, nous avons trouvé beaucoup de données impertinentes. Par exemple. pendant la phase d'exploration nous avons trouvé que le "**Weight**" est une donnée impertinente car elle très corrélée avec le "**BMI**". On a décidé de sauvegarder le BMI et de laisser tomber le **"Weight"**. De même que nous nous sommes

rendu compte que le **"RawDX"** et **"GHOA"** étaient les mêmes données sauvegardées dans d'autres formes. Et on a maintenu le **"RawDX"** et laisser tomber le **"GHOA"**

3.4.4 Données manquantes

Les données manquantes **(Missing data en anglais)** sont un problème trompeur dans l'apprentissage automatique appliqué. Tout d'abord, juste pour être clair, vous ne pouvez pas simplement ignorer les valeurs manquantes dans un ensemble de données. Vous devez les gérer d'une manière ou d'une autre pour la raison très pratique que la plupart des algorithmes n'acceptent pas les valeurs manquantes.

Les deux stratégies les plus couramment recommandées pour traiter les données manquantes sont les suivantes [16] :

1. Suppression des observations qui ont des valeurs manquantes **(Dropping en anglais)**
2. Imputation des valeurs manquantes en fonction d'autres observations **(Imputing en anglais)**

Après avoir essayé les deux recommandations, nous avons constaté que le **"imputing"** nous donne de meilleurs résultats. Pour arriver à cette affirmation, nous avons essayé plusieurs modèles avec leurs différentes évaluations. Il faut souligner que notre dataset a de beaucoup de valeurs nulles. C'est la deuxième difficulté du projet après le déséquilibre au niveau du dataset.

Il est important de préciser que dans le **"imputing"**, il y a plusieurs méthodes de remplissage. On peut remplir avec la valeur "**zéro (0)**", la valeur moyenne "**mean**" ou la valeur moyenne "**median**". Encore, nous avons opté de remplir avec la valeur moyenne "**mean**", car elle nous a donné de meilleurs résultats.

3.5 Feature Engineering

L'ingénierie des caractéristiques **(Feature Engineering en anglais)** consiste à créer de nouvelles entités en entrée à partir de celles qui existent déjà. En général, vous pouvez considérer le nettoyage des données comme un processus de soustraction et l'ingénierie des fonctionnalités comme un processus d'ajout. C'est souvent l'une des tâches les plus précieuses qu'un data scientist puisse faire pour améliorer les performances du modèle, et ce, pour trois grandes raisons [18] :

1. Vous pouvez isoler et mettre en évidence les informations clés, ce qui aide vos algorithmes à se concentrer sur ce qui est important.
2. Vous pouvez apporter votre propre expertise de domaine.
3. Plus important encore, une fois que vous comprenez le «vocabulaire» de l'ingénierie des caractéristiques, vous pouvez apporter l'expertise de domaine d'autres personnes !

Le feature engineering est l'étape qui précède la modélisation. Il prend en entrée le jeux de données brutes et produit en sortie un jeux de données préparé pour l'étape de la modélisation. Le modèle prend en entrée le jeux de données produit par le processus du feature engineering.

Nous avons fait notre feature engineering en deux étapes :

1. Premièrement, les features **Dx1, Dx2, Dx3** qui existent déjà dans notre dataset doivent être combinés entre eux pour créer un nouveau feature appelé **degree_dx** qui est le niveau général diagnostique du patient. Nous soulignons que **Dx1, Dx2, Dx3** représentent respectivement : premier niveau diagnostique, deuxième niveau diagnostique et troisième niveau diagnostique du patient. En combinant ces trois diagnostiques, nous créons un niveau général diagnostique du patient.
   Logique de création : On regroupe ces colonnes (**Dx1, Dx2, Dx3**) suivants la logique si les 3 sont presents (c'est-à-dire si les 3 ont valeurs non-null), on donne la valeur de 3 ; si uniquement DX2 et DX1 sont présents, on donne la valeur de 2 ; si uniquement DX1 on donne la valeur de 1 si aucune valeur n'est présente, on donne la valeur 0. Le feature résultant se nomme **degree_dx**
2. Deuxièment, nous créons un nouvel autre feature appelé **medcond** qui détermine les conditions médicales du patient. Nous nous basons sur l'ensemble de caractéristiques pré-opératoires du patient. Par exemple son niveau de glucose, les quantités d'hémoglobine blanches et d'hémoglobines rouges. Ces features sont au nombre de 46 au total.
   Logique de création : On prend toutes les colonnes de conditions médicales (**PreOpHgb** à **Depression**, ils sont au nombre de 46 dans le dataset) : On donne la valeur 1 lorsqu'on trouve une valeur non-nulle à chacune des conditions sauf pour les colonnes suivantes :
   - On donne la valeur 2 pour les colonnes suivantes : [**PreOpHgb, PreOpGlucose, pulm circ, other neuro, chronic pulm**]
   - On donne la valeur 3 pour les colonnes suivantes [**PreOpC, Paralysis, renal failure, liver failure**]

3.6 Modélisation des données

La modélisation est un processus d'entrainement de modèles. Il consiste à décrire comment choisir le meilleur modèle qui s'adapte à la problématique qu'on étudie et les raisons pour lesquelles le modèle choisi est meilleur. Ici, meilleur modèle ne veut pas dire qu'un algorithme est meilleur qu'un autre mais tout dépend du cas qu'on étudie.

Toutes les étapes que nous venons de détailler dans les sections précédentes constituent des étapes préparatoires pour la modélisation. Dans l'étape d'exploration de données, nous nous sommes rendus compte que notre ensemble de données était déséquilibré.

Puisque notre ensemble de données est déséquilibré, nous ne pouvons pas utiliser les algorithmes de classification simples tel que LogisticRegression, TreeDecisionClassifier, LinearClassifier etc. Nous avons utilisé l'approche des méthodes d'ensemble qui consiste à combiner plusieurs classificateurs simples pour donner un classificateur fort. L'image 3.3 donne une approche générale sur le fonctionnement et la construction des méthodes d'ensemble.

Dans le chapitre sur *"Contexte théorique"* nous avons vu comment manipuler les données déséquilibrées en ré-échantillonnant **(resampling en anglais)** les données d'origine pour fournir des classes équilibrées. Nous avons vu toutes les avantages à utiliser les techniques SMOTE que les autres. Dans la sous-section qui suit sur la modélisation, nous allons brièvement examiner une autre approche, à savoir la modification des algorithmes de classification existants pour les rendre appropriés aux ensembles de données déséquilibrés.

3.6.1 Méthodes d'ensemble

L'objectif principal de la méthodologie d'ensemble **(Ensemble methods en anglais)** est d'améliorer la performance des classificateurs uniques. L'approche consiste à construire plusieurs classificateurs à deux étapes à partir des données originales, puis à agréger leurs prédictions [1]. Les méthodes d'ensemble sont nombreuses, mais nous allons brièvement définis les trois plus importantes : Bagging, Boosting et Random Forest.

3.6.1.1 Bagging

Bagging (Bootstrap aggregating) est une approche de construction d'ensemble qui utilise différents sous-ensembles de données d'apprentissage avec une méthode

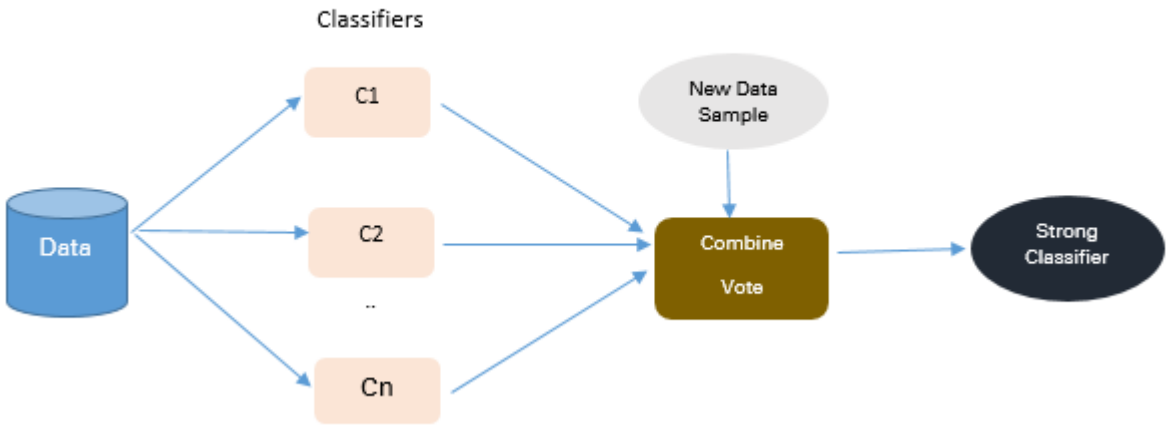

Fig. 3.3 – Approach to Ensemble based Methodologies [22]

de classification unique. Étant donné un ensemble d'apprentissage de taille **t**, Bagging attire des instances aléatoires **t** de l'ensemble de données avec remplacement (à l'aide d'une distribution uniforme). Ces **t** instances sont apprises, et ce processus est répété plusieurs fois. Étant donné que le tirage au sort est effectué avec remplacement, les instances tirées contiendront des doublons et des omissions par rapport à l'ensemble d'apprentissage initial. Chaque cycle à travers le processus aboutit à un classificateur. Après la construction de plusieurs classificateurs, les sorties de chaque classificateur sont combinés pour produire la prédiction finale [28].

3.6.1.2 Boosting

Une autre approche appelée «Boosting» utilise également une méthode d'apprentissage unique avec différents sous-ensembles de données d'apprentissage. Sa structure globale est similaire à celle de la méthode Bagging, à la différence qu'elle conserve la trace de la performance de l'algorithme d'apprentissage et se concentre sur les cas qui ne sont pas correctement appris. Au lieu de choisir les instances **t** d'apprentissage à l'aide d'une distribution uniforme de manière aléatoire, les exemples d'apprentissage sont sélectionnés en favorisant les instances quine sont pas bien classées. Après plusieurs cycles, la prédiction est réalisée selon un vote pondéré des prédictions de chaque classificateur. Ainsi, les poids sont proportionnels à la précision de chaque classificateur sur son ensemble d'apprentissage. L'algorithme le plus connu de l'approche Boosting, appelée « AdaBoost » [28].

3.6.1.3 Stacking

Le stacking est similaire au boosting : vous appliquez également plusieurs modèles à vos données d'origine. La différence ici est, cependant, que vous n'avez pas juste une formule empirique pour votre fonction de poids, plutôt que vous introduisez un méta-niveau et utilisez un autre modèle / approche pour estimer l'entrée avec les sorties de chaque modèle pour estimer les poids ou , en d'autres termes, pour déterminer quels modèles fonctionnent bien et ce qui est mal donné ces données d'entrée [8].

3.6.1.4 Random Forest

Les forêts aléatoires(plus connus sous **Random Forest**) sont une combinaison d'arbres de décision, où chaque arbre dépend des valeurs d'un vecteur aléatoire indépendamment échantillonné et avec la même distribution pour tous les arbres de la forêt.L'erreur de généralisation d'une forêt d'arbres dépend de la force des arbres individuels dans la forêt et de la corrélation entre eux. L'utilisation d'une sélection aléatoire de caractéristiques pour diviser chaque nœud donne des taux d'erreur qui se comparent favorablement à AdaBoost. [1]

Pour notre projet nous avons entrainé quatre algorithmes ensemblistes ; deux d'entre eux sont des algorithmes de boosting, un algorithme stacking et le random forest. La table 3.4 présente une comparaison des différents résultats obtenus en entrainant ces quatre types d'algorithmes ensemblistes différents. Pour sélectionner le meilleur algorithme, nous avons développé des processus automatiques en nous basant sur les scores de chaque modèle. Le meilleur modèle va être celui du meilleur score obtenu.

Algorithm	**Metrics**		
	Accurency score	**Precision**	**Recall**
GradientBoostingClassifier	0.83	0.93	0.89
AdaBoostClassifier	0.85	0.93	0.90
StackingClassifier	0.78	0.91	0.84
RandomForestClassifier	0.82	0.91	0.90

TABLE 3.4 – Comparison of ensemble methods algorithms used in our project

Nous tenons compte aussi des metrics d'évaluation comme "Precision and Recall".Ces derniers nous permettent de déterminer la performance de notre modèle à classifier les classes positives et celles qui sont négatives. Le but de notre projet était d'obtenir la précision en dessus de 90%. Généralement, dans la pratique une

telle précision est acceptable pour pouvoir fier le modèle.

Vous pouvez constater à la table 3.4 que nous obtenons des résultats différents pour chaque algorithme. Pour faire le choix du bon algorithme, notre API machine learning est capable d'évaluer chaque algorithme de façon indépendante et itérativement pour déterminer le meilleur modèle, tout en combinant les différents résultats de chaque metric. Dans ce cas, c'est l'**AdaBoostClassifier** qui serait le modèle estimé meilleur.

Après avoir déterminé le meilleur modèle à partir des processus automatiques. La dernière étape qui nous reste à faire est la sérialisation de notre modèle pour son utilisation future soit dans la phase de déploiement. Parce qu'un modèle doit être capable de prédire de nouvelles données (**online data**). La section qui vient expliquera en détail comment nous avons procédé pour déployer notre modèle c'est-à-dire le rendre accessible à des utilisateurs qui ne connaissent rien en machine learning.

3.7 Déploiement de modèle

Les systèmes modernes d'apprentissage automatique facilitent la construction d'un système de décision de base. Cette facilité, cependant, est un peu décevante. Construire et déployer un premier système de décision a tendance à bien fonctionner, et les résultats peuvent être assez impressionnants pour les bonnes applications. L'ajout d'un autre système se passe généralement aussi bien. Cependant, d'étranges interactions peuvent commencer à apparaître d'une manière qui serait impossible du point de vue de l'ingénierie logicielle. Changer une partie du système affecte une autre partie du système, même si des tests isolés peuvent suggérer que cela est impossible.

Le problème est que les systèmes basés sur l'apprentissage automatique peuvent avoir des propriétés très subtiles qui sont très différentes des systèmes logiciels plus traditionnels. En partie, cette différence vient du fait que les sorties des systèmes d'apprentissage automatique ont des comportements beaucoup plus complexes que les composants logiciels typiques. Cela vient aussi en partie du fait de la nature probabiliste des jugements que de tels systèmes sont appelés à faire.

Cette complexité et cette subtilité rendent la gestion de ces systèmes plus délicate que la gestion de systèmes logiques traditionnels, bien modularisés, basés sur des micro-services. Les systèmes complexes d'apprentissage profond peuvent

évoluer pour montrer des comportements pathologiques «changer quoi que ce soit, tout changer», même s'ils apparaissent superficiellement comme des micro-services bien conçus avec de hauts degrés d'isolation [7].

Compte tenu de cette complexité qui existe au niveau des applications de machine learning, nous avons décidé de construire un API pour faire le déploiement de notre modèle, de telle sorte que s'il faut faire un changement dans la phase d'entrainement du modèle, cela n'affectera pas tout le système en général.

3.7.1 Construction d'un API

Les API (Application Program Interfaces) sont des méthodes de communication logicielle développées sur une norme particulière. De nombreuses entreprises ont leurs propres API publics qui résolvent des problèmes spécifiques pour les développeurs. En transmettant à leur API un paramètre en entrée, l'utilisateur peut recevoir une sortie sans avoir besoin de savoir (ou de comprendre) comment la tâche sous-jacente est effectuée. Les API permettent une fonctionnalité multi-plateforme, rendue possible par une norme agnostique de plate-forme, telle que la spécification REST. La beauté des API est qu'elles sont super accessibles - que vous construisiez une application mobile, des appareils IoT, un serveur ou que vous souhaitiez simplement trouver un moyen de communiquer entre vos propres microservices, les API vous y aident [5].

Pour communiquer avec notre modèle, nous avons construit un API en Django framework web en suivant l'architecture présentée dans la figure 3.4. Dans cette architecture, le model peut être soit un fichier ou une base de données ; mais dans notre cas, le model est notre model machine learning sérialisé depuis dans la phase d'entrainement.

3.7.2 Présentation de notre API

La table 3.5 nous présente les détails de notre API. La colonne "**Attribute or Method**" désigne les différents attributs et les méthodes que constitue notre API. et "**Comments**" dit ce que fait chaque attribut ou méthode. On précise aussi les différents paramètres de chaque fonction et leur valeur de retour.

La construction de l'API est un travail d'ingénierie de logiciels, ce qui permet de combiner ensemble deux champs d'études de l'informatique (**Machine learning et Software Engineering**). Il faut relater que les modèles de machine learning ont besoin d'une couche d'ingénierie de logiciels pour leur rendre accessible aux utilisateurs qui ne comprennent rien en machine learning. Dans la vraie vie, ceux qui utilisent les modèles ne connaissent rien ni en machine learning ni en software

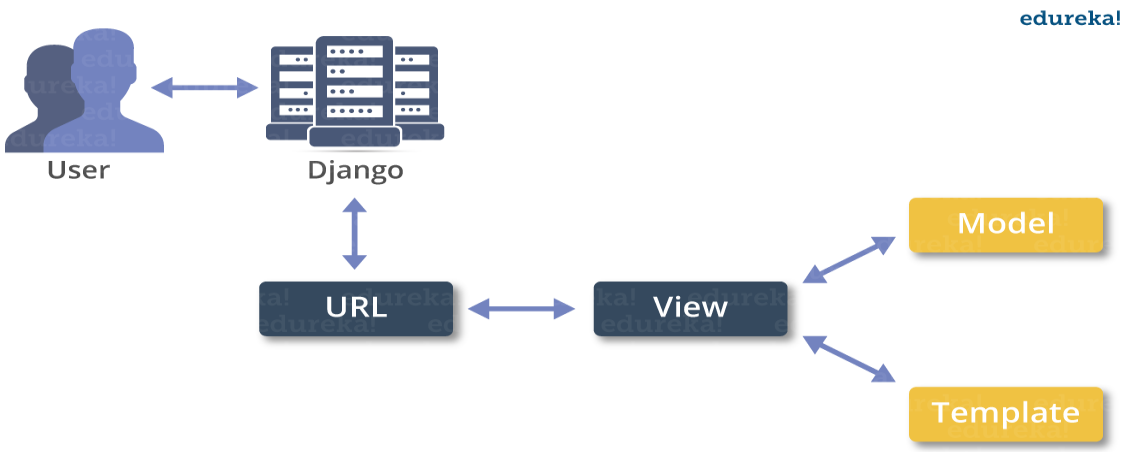

Fig. 3.4 – Django Framework web architecture [11]

engineering. C'est pourquoi dans tout lieu où il y a une équipe de data scientist qui travaillent, il y a aussi une équipe de développement de logiciels.

3.8 Conclusion

Attribute or Method	Comments
model	Variable that allows to recover model saved during the training phase
preprocess_request(form)	Method to prepare the data for transformation. Its loads the yaml file that contains the set of attributes wich are authorized and converts null values in NaN, Yes in 1 and No in 0. It returns a dcitionnary of attributes and values.
predict(form)	Function that allows prediction of new data. It takes a "form" input parameter and returns a decision message that says whether the patient should be fired or left. The "form" parameter is a django web form
save_prediction(form, y_pred)	This function saves the new predictions in a database in order to analyze the behavior of the model in the future. It takes as input two parameters (form, y_pred). The "form" parameter is always the same django form and "y_pred" is the decision made by the model.
get_predictions()	Returns all predictions

TABLE 3.5 – Presents details of our API

4 | Évaluation

L'évaluation des algorithmes d'apprentissage automatique est une partie essentielle de tout projet. Le modèle peut nous donner des résultats satisfaisants lorsqu'il est évalué à l'aide d'une métrique, par exemple **precision_score**, mais il peut donner de mauvais résultats lorsqu'il est évalué par rapport à d'autres mesures, telles que **logarithmic_loss** ou toute autre mesure de ce type. Évaluer un projet de machine learning est différent d'évaluer les projets de génie logiciels traditionnels. La différence est que pour évaluer de tels projets, il faut tenir compte des différentes théories de machine learning permettant de faire des évaluations sur des modèles. Ce que nous avons déjà dans la sous-section 2.3.1 portant sur la "**Mesure de performance des algorithmes de classification**" au chapitre "**Contexte théorique**".

Pour évaluer notre modèle, nous avons séparé notre ensemble de données en deux (Train Set et Test Set). Le train set a été utilisé pour construire le modèle et le test set pour faire l'évaluation du modèle. Nous avons séparé le jeu de données comme suit : 80% pour le train set et 20% pour le test set. Puisque notre jeu de données était déséquilibré, nous n'avons pas ré-échantillonné (resampling en anglais) notre ensemble de test, parce que les nouvelles données que le modèle va prédire se seront jamais ré-échantillonné. D'autres raisons de ce choix est que si nous ré-échantillonnons l'ensemble de test, il est fort probable que le modèle soit biaisé, ce qui provoquera de fausses prédictions.

La table 4.1 donne les résultats de l'évaluation de notre modèle sélectionné comme le meilleur. On pourrait constater que cette table est ressemblée à la table de comparaison des différents algorithmes entrainés dans le projet (table 3.4). La raison est que pour comparer et sélectionner les différents algorithmes, il faut d'abord évaluer chaque algorithme. C'est comme une évaluation anticipée. Une interprétation sommaire de cette évaluation est que 8% des prédictions faites par le modèle est erroné et 92% sont correctement classées. Pour faire cette interpretation, nous nous tenons compte du " **Precision score** " de la table 4.1.

Model's name	AdaBoostClassifier
Precision score	0.9291845493562232
Recall score	0.9002079002079002
Area under Curve	0.8511029411764706
F1 score	0.914466737064414

TABLE 4.1 – Gives different results of our model evaluation

5 | Limitations et travaux futurs

Notre projet est loin d'être exhaustif, il reste beaucoup de choses à faire mais les parties essentielles du projet sont réalisées. Nous pouvons considérer tout ce que nous avons fait jusqu'à présent dans ce projet comme étant une première version du système. Pour le moment, on peut utiliser le système pour faire la prédiction et l'évaluation de notre modèle déployé. C'est-à-dire n'importe quel orthopédiste peut utiliser le système pour prédire si son patient doit rester à l'hôpital et s'il doit retourner chez lui après une arthroplastie. Pour que le médecin orthopédiste puisse faire ces prédictions, il doit, préalablement remplir un formulaire web qui contient les champs nécessaires pour compléter un dossier médical.

Comme travaux futurs, nous prévoyons inclure les modules qui gèrent le monitoring et management du modèle. Et aussi la gestion des sessions puisque tout le monde peut avoir accès au module de prédictions mais les modules monitoring et management sont réservés aux administrateurs du système.

6 | Travaux connexes

Nous n'avons pas trouvé de travaux connexes directement liés à notre domaine c'est-à-dire aucun organisme n'a encore développé de systèmes liés au risque de retour des patients après une arthroplastie. Par contre nous avons trouvé des projets qui ont des points similaires par rapport à notre projet tels que les classes déséquilibrées.

Philip K. Chan, Computer Science, Florida Institute of Technology & Salvatore J. Stolfo, Department of Computer Science, Columbia University [3] ont travaillé sur un projet de détection de fraude par carte de crédit. Ils ont eu leur dataset déséquilibrés (distributions uniformes dans leur article), 20% de cas étaient frauduleux contre 80% non-frauduleux.

7 | Conclusion générale

Bibliographie

[1] D. Amina and D. Soumia. Etude comparative des methodes ensemblistes de classification des donnes medicales. *Universite Abou Bakr Belkaid de Tlemcen*, 2017.

[2] B. Bashinskaya, R. M. Zimmerman, B. P. Walcott, and V. Antoci. Arthroplasty utilization in the united states is predicted by age-specific population groups. *ISRN Orthopedics*, 2012.

[3] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions : A case study in credit card fraud detection. *American Association for Artificial Intelligence*, 1998.

[4] W.-L. Chao. Machine learning tutorial. Technical report, DISP Lab, Graduate Institute of Communication Engineering, National Taiwan University, 2011.

[5] Datmo. Deploying a machine learning model as an api with datmo, falcon, gunicorn, and python. https ://blog.datmo.io/deploying-a-machine-learning-model-as-an-api-with-datmo-falcon-gunicorn-and-python-1e422fa5fa49, 2012.

[6] G. developers. Classification : Roc et auc. https ://developers.google.com/machine-learning/crash-course/classification/roc-and-auc, 2008.

[7] T. Dunning and E. Friedman. *Machine Learning Logistics*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2017.

[8] S. Exchange. Bagging, boosting and stacking in machine learning. https ://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning, November 2012.

[9] M. A. Galvis-Narinos F. Le système de santé des états-unis. *Pratiques et Organisation des Soins*, 2009.

[10] https ://www.europusa.com. Comment fonctionne le système de santé et les assurances aux etats-unis ? https ://www.europusa.com/assurance/pourquoi-et-comment-choisir-une-bonne-assurance-sante/comment-fonctionne-le-systeme-de-sante-et-les-assurances-aux-etats-unis/, 2007.

[11] A. Johari. Django tutorial – web development with python django framework. https ://www.edureka.co/blog/django-tutorial/, September 2017.

[12] F. Jonah Hebert-Davies, M.D. Biography of jonah hebert-davies. http ://www.orthop.washington.edu/ ?q=faculty-profiles/jonah-hebert-davies-md-frcsc.html, 2013.

[13] C. J. Lavernia, V. H. Hernandez, and M. D. Rossi. Payment analysis of total hip replacement. *Centers for Medicare and Medicaid Services*, pages http ://www.larkinhospital.com/larkinorthopedics/wp–content/uploads/2014/01/Payment–Analysis–of–Total–Hip–Replacement1.pdf, 2007.

[14] W. J. H. Michael Masaracchio and al. Timing of rehabilitation on length of stay and cost in patients with hip or knee joint arthroplasty : A systematic review with metaanalysis. *PLOS ONE*, 2017.

[15] Openclassrooms. Initiez-vous au machine learning. https ://openclassrooms.com/courses/initiez-vous-au-machine-learning/comment-resoudre-un-probleme-de-data-science, 2014.

[16] E. D. Science. Data cleaning. https ://elitedatascience.com/data-cleaning, 2011.

[17] E. D. Science. Exploratory analysis. https ://elitedatascience.com/exploratory-analysis, 2011.

[18] E. D. Science. Feature engineering. https ://elitedatascience.com/feature-engineering, 2011.

[19] M. Sunasra. Performance metrics for classification problems in machine learning. https ://medium.com/greyatom/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b, 2011.

[20] M. J. Towler. Confusion matrix. https ://alearningaday.com/2016/09/14/confusion-matrix/, September 2014.

[21] A. R. O. M. L. TRENDS and F. PROSPECTS. Manish kumar aery & chet ram. *Research Cell : An International Journal of Engineering Sciences*, 25(63019) :http ://ijoes.vidyapublications.com/, November 2017.

[22] A. Vidhya. How to handle imbalanced classification problems in machine learning ? https ://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/, 2011.

[23] Wikipedia. Apprentissage automatique. https ://fr.wikipedia.org/wiki/Apprentissage_automat 2011.

[24] Wikipedia. Apprentissage supervisé. https ://fr.wikipedia.org/wiki/Apprentissage_supervis

[25] Wikipedia. Arthrose. https ://fr.wikipedia.org/wiki/Arthrose, 2011.

[26] Wikipedia. Data science. https ://wikipedia.org/wiki/Data_science, 2011.

[27] Wikipedia. Statistical classification. https ://en.wikipedia.org/wiki/Statistical_classification, 2011.

[28] H. K. Zouari. *Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation.* Laboratoire PSI - FRE CNRS 2645, Décembre 2004.