Stage de Recherche
Master 2 MVA de l'ENS Cachan

Nouvelles techniques de Machine Learning pour la médecine :

Facteurs prédictifs de la réhospitalisation précoce de drépanocytaires adultes pour crise vaso-occlusive

Simon Bussy
simon.bussy@gmail.com

encadré par
Anne Sophie Jannot, **annesophie.jannot@aphp.fr**
Stéphane Gaiffas, **stephane.gaiffas@cmap.polytechnique.fr**
Agathe Guilloux, **agathe.guilloux@upmc.fr**

Du 15 Avril au 15 Septembre 2015

1. Introduction

Ce stage de recherche est effectué dans le cadre du Master 2 Mathématiques Vision et Apprentissage de l'ENS Cachan, conjointement au sein du laboratoire du Centre de Mathématiques Appliquées de l'école Polytechnique (CMAP) et de l'équipe INSERM n° 22 basée à l'Hôpital Européen Georges Pompidou (HEGP) ; et financé par l'Institut Europlace de Finance.

L'HEGP est un centre de référence pour le traitement des Crises Vaso-Occlusives (CVO) dont souffrent certains patients atteints par la drépanocytose. Ces crises se caractérisent par des douleurs osseuses intenses dues à la falciformation des hématies, qui peuvent nécessiter un traitement morphinique intraveineux et une hospitalisation de quelques jours pour l'administrer.

Il n'existe pas d'élément clinique objectif ou de paramètre biologique validé permettant de déterminer lorsqu'une CVO est terminée, et donc que le patient peut quitter l'hôpital en toute sécurité. Le critère principal de sortie d'un patient hospitalisé pour CVO est l'évaluation de sa douleur, réalisée sous traitement antalgique majeur. Les sorties prématurées sont responsables de réadmissions précoces via les urgences et de prolongations du séjour hospitalier total. Le taux de réhospitalisation étant élevé (environ 20%), l'idée générale du stage est de se baser sur l'ensemble des données observées lors des séjours des drépanocytaires afin de construire un modèle de prédiction des réhospitalisations précoces.

Table des matières

2. Remerciements

Je tiens ici à remercier chaleureusement mes trois encadrants de stage avec qui j'ai passé cinq mois très enrichissants. Tout d'abord, Anne-Sophie Jannot qui m'a accueilli au sein de l'HEGP et qui a toujours été très disponible pour m'aider à comprendre les données et interpréter les résultats. Puis Agathe Guilloux et Stéphane Gaiffas avec qui j'ai progressé d'une part techniquement, puisque j'étais novice en python au début du stage, et d'autre part théoriquement avec la mise en pratique de certaines méthodes nouvelles pour moi. Le déroulement du stage laisse ainsi présager une belle collaboration lors de mon futur doctorat que je débuterai en octobre prochain entouré de ce même trio de chercheurs.
Je tiens également à remercier mon camarade stagiaire Samy Jelassi sans qui les pauses café n'auraient pas eu la même saveur.

3. Présentation générale

3.1. Contexte du stage

Le projet repose sur la complémentarité entre l'équipe de statisticiens du CMAP et l'équipe INSERM de modélisation en santé "Science de l'Information au service de la Médecine Personnalisée" de l'HEGP ; hôpital de l'AP-HP (Assistance Publique – Hôpitaux de Paris).

Le CMAP est déjà très actif dans le domaine des statistiques appliquées au monde médical, et plus particulièrement l'équipe Signal IMage Probabilités numériques Apprentissage Statistique (SIMPAS). Cette équipe regroupe des chercheurs dans le domaine de l'aléatoire au sens large, dont les travaux sont axés sur le traitement numérique des données ou des modèles aléatoires, allant des fondements théoriques des algorithmes et méthodes, aux développements informatiques efficaces.

Le service d'accueil au sein de l'HEGP est le Service d'Informatique Médicale, dont les activités sont localisées géographiquement à l'HEGP dans le 15ème arrondissement à Paris et au Centre de Recherche des Cordeliers dans le 6ème. L'AP-HP, est le Centre hospitalo-universitaire (CHU) d'Île-de-France et le 1er CHU d'Europe. Les 92 000 professionnels qui y sont ratachés s'engagent à offrir des soins de grande qualité 24h/24. Avec 37 hôpitaux, réunis en 12 groupes hospitaliers et une fédération du polyhandicap, l'AP-HP propose, en lien étroit avec les acteurs sanitaires et médico-sociaux de la région, des soins et des modes de prise en charge adaptés aux besoins de santé de proximité. Elle met également toute son expertise et ses capacités d'innovation au service des patients atteints de pathologies complexes. Grâce à l'association du soin, de l'enseignement et de la recherche, les 7 millions de personnes soignées chaque année par des équipes de renommée internationale bénéficient de traitements de pointe dans l'ensemble des disciplines médicales.

L'HEGP dispose d'un entrepôt de données cliniques de 1,4 million de patients avec par exemple plus de 100 millions de résultats d'examens biologiques et le plus long historique en France (15 ans). En effet, depuis l'ouverture de l'hôpital en 2000, chaque patient dispose d'un dossier informatisé [14]. La base complète de l'entrepôt de données cliniques de l'HEGP contient près de 700 000 patients différents. Les données disponibles couvrent la totalité du dossier patient (consultations et hospitalisations) : codage des diagnostics et des actes, comptes rendus textuels, suivi des paramètres vitaux (tension, poids, etc), ensemble des résultats biologiques, prescriptions médicamenteuses. Parmi les valeurs biologiques collectées dans cette base de données figurent des variables non exploitées par les praticiens. En effet, les analyses biologiques prescrites par les praticiens à la recherche d'une orientation diagnostique donnent parfois des résultats inattendus qui ne peuvent pas être exploités du fait de l'absence de données scientifiques pour les utiliser. Dans l'exploitation de cet entrepôt, on rencontre des problématiques de choix de seuils notamment dans deux domaines : l'aide au diagnostic et le monitoring de la durée d'hospitalisation [9].

L'équipe INSERM à laquelle je suis ratacché (Equipe 22 du Centre de Recherche des Cordeliers, dirigée par Anita Burgun) a pour objectif le développement d'approches associant des méthodes informatiques et statistiques pour combiner des données cliniques et génétiques dans le but d'offrir une médecine "personnalisée", c'est-à-dire pouvoir proposer des parcours de soins adaptés aux caractéristiques cliniques et génétiques de chacun.

3.2. Objectifs visés

Le projet développé au cours de ce stage se concentre donc sur une pathologie bien particulière : la drépanocythose. Comme évoqué dans l'introduction, cette maladie engendre des Crive Vaso-Occlusive (CVO) qui nécéssitent en général un séjour à l'hôpital pour soulager les douleurs induites. Les cliniciens considèrent qu'il y une réadmisison précoce dès lors qu'un patient retourne à l'hôpital pour une CVO dans les 15 jours suivant la sortie d'un séjour similaire, et cela constituera l'une des hypothèse de travail de base. Actuellement, le taux de réadmission précoce après une CVO est de l'ordre de 17% à l'HEGP. Ce taux est comparable à celui obtenu sur une analyse de plus de 10000 séjours pour CVO aux Etats-Unis, qui montrait un taux de 22%, et ce dernier est plus fréquent chez les patients jeunes [5].
Ce taux est donc élevé et le diminuer permettrait d'améliorer la qualité des soins pour différentes raisons. Il a par exemple été montré que des réadmissions précoces étaient à l'origine d'une morbimortalité accrue [2]. La réadmission via les urgences est souvent traumatisante pour ces patients hyperalgiques pour lesquels le retard de prise en charge antalgique est un facteur de mauvais pronostic, qui allonge la durée d'hospitalisation globale. Mais peu de données existent sur les facteurs de risque modifiables de réadmission.

A l'HEGP, le dossier patient informatisé comprend toutes les données structurées et codées (observation médicale, prescriptions, signes vitaux, examens biologiques et radiologiques etc). Le clinicien a donc actuellement à sa disposition de très nombreux marqueurs biologiques pour l'aider dans sa démarche diagnostique. Mais l'analyse conjointe de marqueurs biologiques pose des problèmes évidents lorsque leur nombre devient très important et qu'il s'agit d'établir une combinaison de marqueurs ayant les meilleures performances diagnostiques.
Les études menées jusqu'à maintenant sur les facteurs impliqués dans les réadmissions précoces se sont concentrées sur des caractéristiques non modifiables des patients, alors que nous nous intéresserons ici à la cinétique de la douleur et des différentes variables biologiques modifiées par la CVO, elles mêmes liées à la cinétique de décroissance du traitement morphinique, facteur contrôlable par le clinicien. Cette étude pourra donc avoir un impact direct sur l'amélioration de la prise en charge des patients hospitalisés pour CVO.

Pour répondre à cette problématique, nous tenterons d'utiliser des méthodes adaptées aux données longitudinales fortement corrélées et de grande dimension. D'un point de vue statistique, les objectifs du stage sont donc multiples :

1. Extraction des données pertinentes depuis l'entrepôt de données.
2. Structuration des données recueillies.
3. Modélisation de l'information temporelle pour les données longitudinales.
4. Développement de techniques d'apprentissage statistique récentes, comme des techniques d'apprentissage supervisé basées sur des modèles de durée ou des modèles logistiques.

Il s'agira par exemple d'utiliser des algorithmes de pénalisation récents sur des modèles statistiques pour données cliniques, et d'appliquer les outils d'optimisation convexe adéquat, sur ce grand volume de données à dimension élevée.

D'un point de vue plus applicatif, les objectifs sont également de taille. La mise en place à terme d'un protocole de décroissance des doses permettrait d'uniformiser les pratiques vers des pratiques fondées

sur les preuves et ainsi d'améliorer les soins et la communication avec ces patients. L'utilisation de ce protocole pourrait être testé dans d'autres centres hospitaliers, recevant des patients avec des caractéristiques semblables. L'impact clinique pourra être important, notamment pour aider à construire un outil d'aide à la décision de fin de CVO regroupant les patients en trajectoires homogènes.

Enfin, cette étude pourra être un accélérateur afin de mettre en place des outils d'aide à la décision dans toutes les situations cliniques où les réadmissions précoces non programmées sont fréquentes. Concernant la perspective médico-économique, une estimation du nombre d'hospitalisations évitables si la décroissance de la cinétique des morphiniques est optimisée, ainsi que du gain associé en utilisant les données du programme médicalisé des systèmes d'information pourra être effectuée.

3.3. Présentation des données

Les données proviennent donc de l'entrepôt de donnée I2B2 de l'HEGP, I2B2 pour "Informatics for Integrating Biology and the Bedside". L'idée derrière la mise en place de cet entrepôt de données est de créer un système de fouille de données capable d'analyser les dossiers médicaux de nombreux patients conservés par les hôpitaux locaux, de recouper les informations et de dégager des liens entre les observations et les pathologies des patients. Le but de ce système est de réussir à extraire des données exploitables de fichiers qui n'ont pas été conçus pour la recherche. La figure 46 en annexe renseigne sur l'organisation des données de l'HEGP, et la figure 47 fournit le diagramme de classes de l'entrepôt.

Les données sont multi-dimensionnelles, évoluent dans le temps et à granularité sémantique fine, ce qui rend leur exploitation directe difficile sans passer par des ontologies. Une autre particularité réside dans le fait que la trajectoire de chaque patient au sein de l'hôpital est unique : si on prend deux patients quelconques, il auront nécessairement subi un certain nombre de tests biologiques différents ; et pour les concepts enregistrés identiques, les instants d'enregistrement des variables considérées ne seront pas les mêmes, ni leur nombre qui dépendra de l'état du patient et de la durée de son séjour. Cela implique une quantité énorme de données manquantes si on considère une structuration classique, où on souhaite créer des features renseignées pour chacun des exemples de notre ensemble d'apprentissage. Chacune de ces difficultés sera discutée par la suite et les choix de modélisation pour y palier seront présentés.

De nombreuses extractions ont été nécessaires pour aboutir à un jeu de données final comportant l'ensemble des informations désirées. Pour être plus clair, les données ont été organisées selon trois grandes catégories : les données dites "basiques", les données biologiques et enfin les données concernant les paramètres vitaux. L'ensemble des données concernent la cohorte de drépanocytaires de l'HEGP entre 2009 et 2015 (avant 2009, les traitements et les protocoles étaient différents).

— Les données "basiques" : il s'agit là des informations démographiques comme le sexe et l'âge, puis la durée du séjour, une variable binaire renseignant le passage ou non du patient en réanimation au cours de celui-ci, et enfin une variable "previous_visit" qui renseigne sur le délais entre ce séjour et le séjour pour CVO précédant. Nous verrons alors par la suite comment nous avons traité le cas des patients ayant un unique séjour, et comment renseigner cette variable pour chaque premier séjour des patients.
— Les données biologiques : il s'agit des résultats de différents tests biologiques réalisés lors du séjour du patient. Les variables retenues après nettoyage sont au nombre de 36 et sont énumérées ci-après : Teneur Globulaire Moyenne, Hémoglobine, Hématies, Leucocytes, Volume Globulaire Moyen, Hématocrite, Nombre de lymphocytes, Plaquettes, Polynucleaires neutrophiles, Lymphocytes, Nombre de monocytes, Monocytes, Polynucleaires eosinophiles, Nombre de polynucleaires neutrophiles, Nombre de polynucleaires basophiles, Nombre de polynucleaires eosinophiles, Polynucleaires basophiles, Chlo-

rures, Cratinine standardise, Sodium, CO_2 Total, DFG estim, Volume Plaquettaire Moyen, Potassium, Protéines, Réticulocytes, ALAT, Phosphatases alcalines, Bilirubine totale, ASAT, Protéine C-reactive, Gamma GT, Erythroblastes, Bilirubine conjugue, Calcium et enfin Glucose.

— Les paramètres vitaux : Comme les données biologiques, les paramètres vitaux sont des données longitudinales enregistrées au cours du séjour, mais la fréquence d'enregistrement est beaucoup plus importante. Les concepts concernés par cette catégorie sont au nombre de 9 et sont énumérés ci-après, suivis de leurs unités respectives et de quelques précisions si besoin. On aura la pression artérielle systolique [mmHg], notée PA max et calculée au moment de la contraction du cœur ; la pression artérielle diastolique [mmHg], notée PA min et calculée au moment de la relaxation du cœur ; la saturation en oxygène [%], test effectué au bout d'un doigt du patient et qui mesure le taux d'oxygène contenu dans les globules rouges après leur passage dans les poumons par rapport à l'hémoglobine ; l'évaluation de la douleur, notée Douleur EVA [U] qui prend une valeur entre 0 et 10 ; la fréquence respiratoire [mvt/min] ; la température [°C] ; le poids [kg] ; l'oxygène [L/min], correspondant au débit d'oxygène administré au patient via un masque de façon à augmenter la saturation ; et enfin la fréquence cardiaque [bpm].

Les choix dans la récupération de toutes ces données ont été fait avec l'aide des cliniciens spécialistes et avec l'expertise de ma tutrice à l'HEGP. De nombreuses difficultés sont apparues après les premières extractions. Pour n'en citer que quelques unes, l'information concernant les paramètres vitaux a par exemple dû être regroupée car présente dans différentes bases : soit codée selon des concepts de paramètres vitaux, soit codée dans une base appelée "pancarte" et correspondant à des questionnaires remplis par les médecins ou infirmiers. Certaines variables n'étaient alors présentes que dans l'une ou l'autre des bases "pancarte" ou "paramètres vitaux" et d'autres dans les deux à la fois, à des intervalles de temps identiques ou non et avec des unités identiques ou non. Un autre exemple serait celui des tris nécessaires pour regrouper les variables biologiques reflétant le même concept mais codées différemment, correspondant par exemple à des quantités physiques identiques mais provenant de tests biologiques différents (l'un sanguin et l'autre urinaire, etc.).

Finalement, les données brutes après la première phase de tri et de nettoyage peuvent se résumer par la figure suivante qui renseigne aussi sur la quantité que représente le jeu de données brutes final.

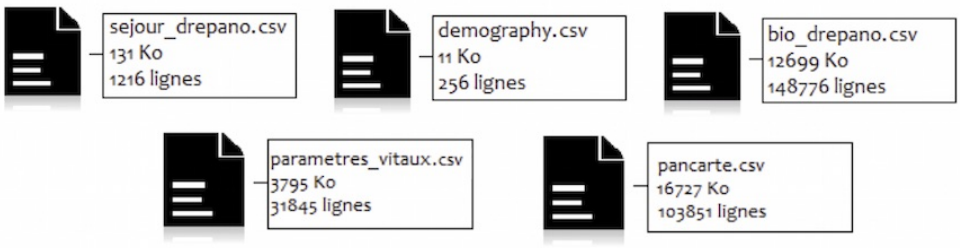

FIGURE 1: Fichiers initiaux des données brutes

Remarque 1 : Je n'ai en fait eu accés qu'à ces fichiers .csv car je n'étais pas autorisé à accéder directement à l'entrepôt I2B2 pour des raisons de protection des données. Les extractions ont été effectuées par ma tutrice à l'HEGP.

Le fichier sejour_drepano.csv contient les informations de base des séjours comme les dates d'entrée et de sortie de l'hôpital, et correspond à la table Visit_drepano dans le diagramme des classes qui suit. Le fichier demography.csv correspond quant à lui aux données dites "basiques" précédemment évoquées. Les fichiers parametres_vitaux.csv et pancarte.csv contenant le même type d'information ont été réorganisés dans une

table unique appelée Vital_parameters dans la figure suivante.

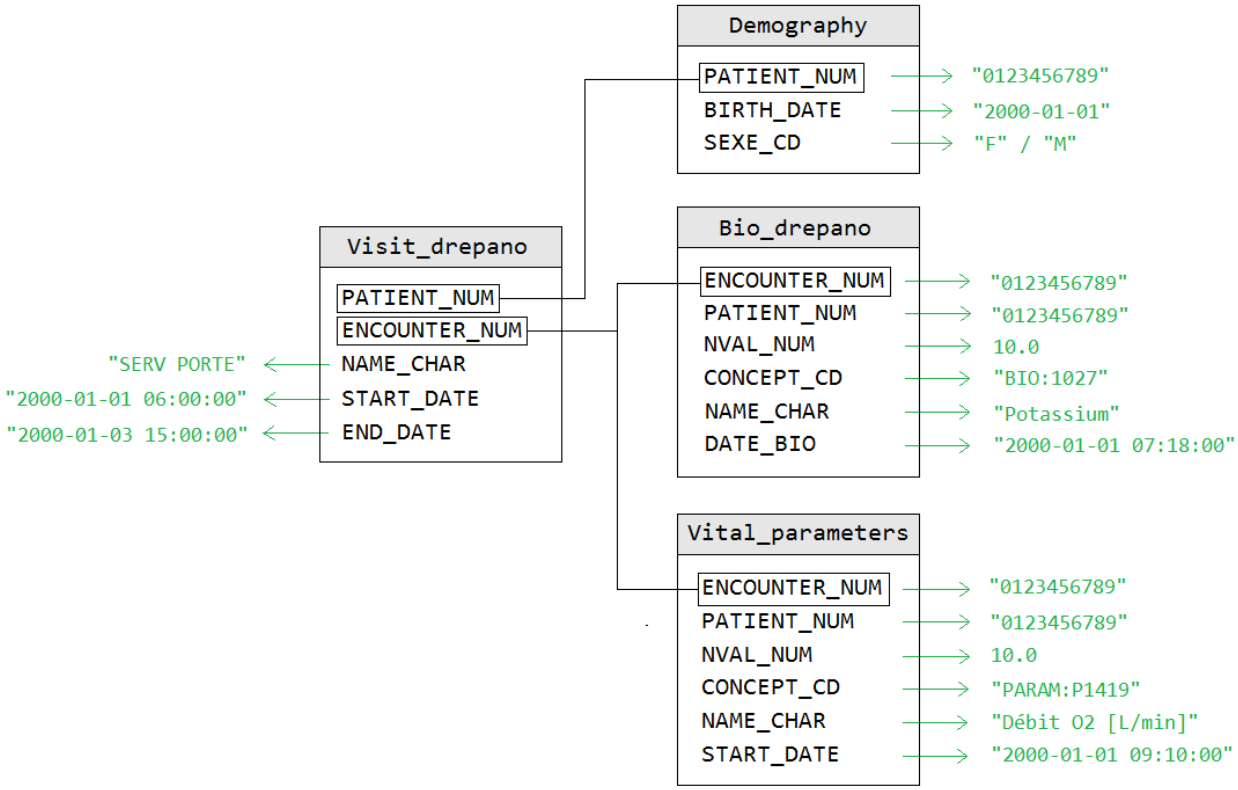

FIGURE 2: Diagramme des classes après le premier nettoyage

4. Création de l'espace de travail

Après ce premier travail d'extraction et de nettoyage des données, la création d'une structure organisant au mieux les informations par patient et par séjour était indispensable avant de pouvoir utiliser et interroger les données de façon efficace.

4.1. Structuration des données

Les données ont alors été organisées dans un fichier JSON dont la structure est explicitée sur la figure suivante. Toutes les informations sont ainsi triées par patient puis par séjour. L'avantage du format JSON est qu'il est facilement compréhensible, la syntaxe n'utilise que quelques marques de ponctuation et il ne dépend d'aucun langage. Comme ce format est très ouvert, il est pris en charge par de nombreux langages et permet de stocker des données de différents types : chaînes de caractères, nombres, tableaux, objets, ou encore booléens ; donc il est tout à fait adapté à nos données. Sa structure en arborescence et sa syntaxe simple lui permet de rester très "léger" et efficace.

```
Patients = {
        '1' = {
                'patient_num' = '0123456789',
                'sex' = 'F/M',
                'birth_date' = '2000-01-01',
                'visits' = {
                        '1' = {
                                'encounter_num' = '0123456789',
                                'age' = '25',
                                'rea' = '0/1',
                                'duration' = '3',
                                'previous_visit' = 'none/10',
                                'next_visit' = 'none/15',
                                'details' = {
                                        '1' = {
                                                'service' = 'SERV PORTE',
                                                'start_date' = '2000-01-01 05:00:00',
                                                'end_date' = '2000-05-01 10:00:00'
                                                },
                                        '2' = {...}
                                        },
                                'bio-infos' = {
                                        '1' = {
                                                'name_char' = 'Potassium',
                                                'val' = {
                                                        '1' = {
                                                                'nval_num' = '10.0',
                                                                'concept_cd' = 'BIO:1027',
                                                                'date_bio' = '2000-01-01 11:00:00'
                                                                }
                                                        '2' = {...}
                                                        }
                                                }
                                        '2' = {...}
                                        },
                                'vital_parameters' = {
                                        '1' = {
                                                'name_char' = 'Débit O2 [L/min]',
                                                'val' = {
                                                        '1' = {
                                                                'nval_num' = '10.0',
                                                                'concept_cd' = 'PARAM:P1419',
                                                                'date_bio' = '2000-01-01 10:30:00'
                                                                }
                                                        '2' = {...}
                                                        }
                                                }
                                        '2' = {...}
                                        }
                                }
                        '2' = {...}
                        }
                },
        '2' = {...}
        }
```

FIGURE 3: Structure du fichier JSON

Le fichier final et unique résultant de cette stucturation est synthétisé par la figure suivante.

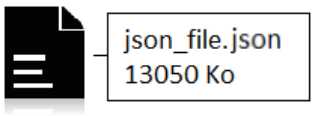

FIGURE 4: Fichier JSON final

Les premières statistiques descriptives des données vont désormais pouvoir être exibées de manière efficace.

4.2. Statistiques basiques sur les données

Commençons par préciser que le nombre de patients différents restant dans le fichier JSON final est de 218, et que le nombre total de visites est de 479. Les premières distributions que l'on souhaite naturellement consulter sont alors la répartition du nombre de visites par patient, et celle du temps inter-visite pour tous les séjours suivis au moins d'un autre séjour (261 en tout).

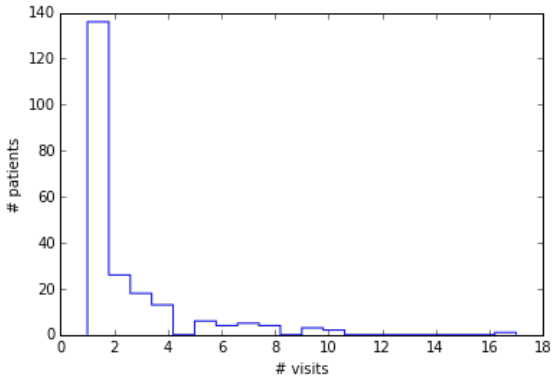

FIGURE 5: Nombre de visites par patient

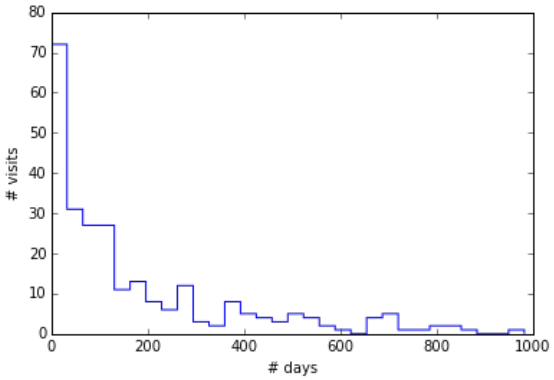

FIGURE 6: Temps inter-visite

Figure	Moyenne	Ecart type	Min	25%	50%	75%	Max
5	2.2	2.1	1.0	1.0	1.0	3.0	17.0
6	180.6	212.2	0.0	28.0	100.0	264.0	983.0

TABLEAU 1 : Statistiques élémentaires pour les figures 5 et 6

Ainsi de nombreux patients n'ont qu'une seule visite dans l'intervalle de temps que nous considérons. En prenant l'ensemble des séjours de la base de données, le taux de rechute dans les 15 jours suivant la sortie et de 10.02 %, ce qui est élevé au vu du nombre de patient à séjour unique. Dans la suite, on appelera "cadre d'apprentissage supervisé" le cadre de prédiction binaire où la variable à prédire vaut 1 si le prochain séjour pour CVO est dans les 15 jours suivant la sortie et 0 sinon.

Les données biologiques :

Visualisons les répartitions élémentaires concernant les variables biologiques ; à commencer par la répartition du nombre de variables biologiques par séjour, puis la distribution du nombre de points enregistrés par variable biologique et par séjour.

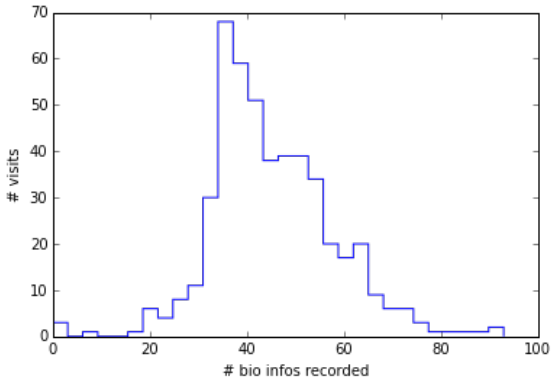

FIGURE 7: Nombre de variables biologiques par séjour

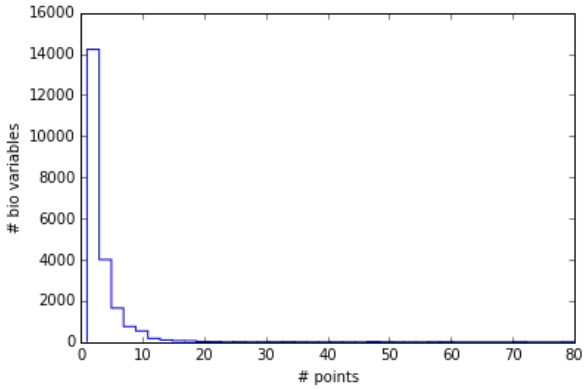

FIGURE 8: Nombre de points par variable biologique

Figure	Moyenne	Ecart type	Min	25%	50%	75%	Max
7	45.3	12.6	0.0	37.0	43.0	53.0	93.0
8	2.9	4.2	1.0	1.0	2.0	3.0	80.0

TABLEAU 2 : Statistiques élémentaires pour les figures 7 et 8

Ainsi, il y a en moyenne très peu de points concernant les signaux biologiques lors des séjours, avec les 3/4 des variables sur l'ensemble des séjours comportant au plus 3 points. Il paraît alors assez difficile d'inférer une quelconque cinétique pour ces données, c'est pourquoi nous nous concentrerons uniquement sur les moyennes des valeurs prises au cours des séjours pour les variables biologiques.

Si on regarde maintenant la proportion de variables biologiques partagées par l'ensemble des visites, c'est-à-dire dont au moins un enregistrement est présent au cours des séjours, on obtient le graphe suivant.

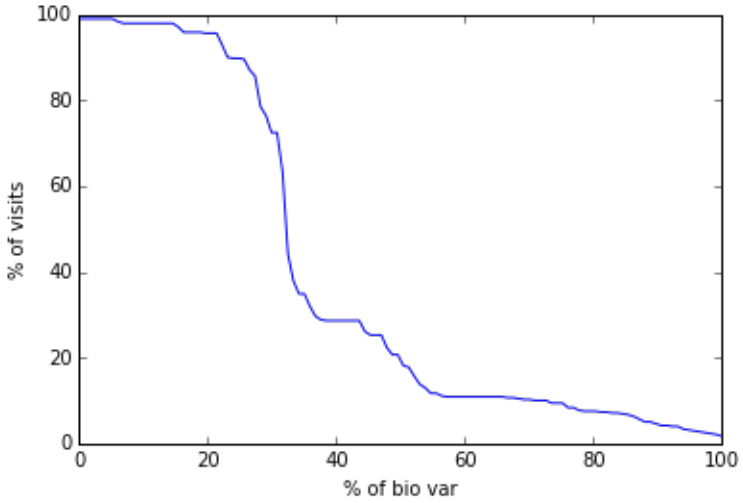

FIGURE 9: Variables biologiques en commun à tous les séjours

Afin d'éviter d'avoir trop de données manquantes, seules les variables biologiques présentes pour 75% des séjours au moins ont été gardées pour la suite, et c'est en fait de cette façon que les 36 variables énumérées dans la section "présentation des données" ont été sélectionnées.

Enfin, pour avoir une idée préalable de l'influence de chaque variable biologique sur la classe binaire dans le cadre d'apprentissage supervisé, effectuons des tests de la somme des rangs de Mann-Whitney-Wilcoxon. Ce test est non paramétrique et permet de tester l'hypothèse selon laquelle la distribution des données est la même dans deux groupes.

Pour être un peu plus précis, si on prend deux échantillons $(X_1^1, \dots, X_{n_1}^1)$ et $(X_1^2, \dots, X_{n_2}^2)$ supposés *i.i.d.* de loi inconnues $\mathbb{P}_1$ et $\mathbb{P}_2$ respectivement, alors l'hypothèse nulle du test est que les deux lois sont identiques.

Soit donc

$$\mathcal{H}_0 \ : \ \mathbb{P}_1 = \mathbb{P}_2$$

L'idée du test est alors la suivante : si on rassemble les deux échantillons, et que l'on range les valeurs dans l'ordre, l'alternance des X_i^1 et des X_j^2 devrait être assez régulière. On aura des doutes sur $\mathcal{H}_0$ si les X_j^2 sont plutôt plus grands que les X_i^1, ou plus petits, ou plus fréquents dans une certaine plage de valeurs. On commence donc par écrire les statistiques d'ordre de l'échantillon global (s'il y a des ex-æquo, on tire au hasard une permutation). On obtient ainsi une suite mélangée des X_i^1 et des X_j^2. On calcule ensuite la somme des rangs des X_i^1, notée W_1 (c'est la statistique de Wilcoxon). Sous l'hypothèse $\mathcal{H}_0$, la loi de W_1 se calcule facilement : sur un échantillon de taille $n_1 + n_2$, il y a $(n_1 + n_2)!$ ordres possibles. Le nombre de rangements possibles des X_i^1 est $\binom{n_1+n_2}{n_1}$, et ils sont équiprobables.

On a donc

$$\forall m \in \left[\!\left[\binom{n_1}{2}, \binom{n_1+n_2}{2} - \binom{n_2}{2} \right]\!\right], \ \mathbb{P}_{\mathcal{H}_0}[\, W_1 = m \,] = \frac{k_m}{\binom{n_1+n_2}{n_1}}$$

où k_m désigne le nombre de n_1-uplets d'entiers $r_1, \ldots, r_{n_1}$ dont la somme vaut m et qui sont tels que

$$1 \leq r_1 < r_2 < \cdots < r_{n_1} \leq n_1 + n_2 \ .$$

Pour les grandes valeurs, on dispose du résultat d'approximation normale suivant :

$$\text{sous l'hypothèse } \mathcal{H}_0, \text{ la loi de } \frac{W_1 - n_1(n_1+n_2+1)/2}{\sqrt{n_1 n_2 (n_1+n_2+1)/12}} \text{ converge vers } \mathcal{N}(0,1).$$

Le score que nous prendrons en compte en sortie du test sera la *p-value*, qui est la probabilité d'observer une statistique de test plus extrême que la réalisation (à partir de l'échantillon) observée sous l'hypothèse nulle. Autrement dit, la *p-value* est le plus petit niveau de significativité pour lequel l'hypothèse nulle aurait été rejetée au vue des données observées. Les résultats des tests sont résumés dans le graphe suivant.

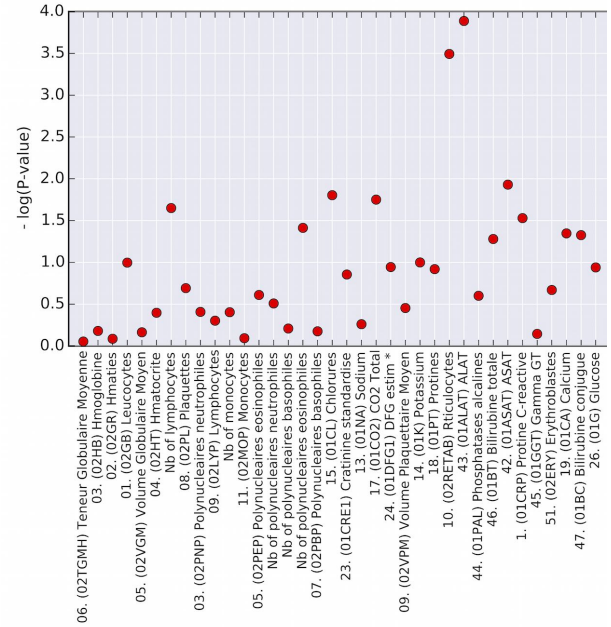

FIGURE 10: Test de Mann-Whitney-Wilcoxons pour les variables biologiques suivant la classe des séjours

De l'information semble donc bien présente dans ces moyennes des résultats de test biologiques au cours des séjours. Deux variables en particulier émergent de ce test : les Réticulocytes et ALAT.
D'un point de vue clinique, l'émergence de ces deux variables est tout à fait cohérente. En effet, les ALAT sont un marqueur de cytolyse hépatique et la cytolyse hépatique fait partie du tableau d'un syndrome majeur drépanocytaire. Les Réticulocytes sont les précurseurs de l'hémoglobine et les valeurs élevées sont un marqueur des syndromes drépanocytaires majeurs du fait de la nécessité d'une régénération importante des globules rouges (érythrocytes).

Les paramètres vitaux :

Observons maintenant le même type de répartition de base concernant les paramètres vitaux, à savoir la répartition du nombre de points par paramétre vital et par séjour, puis la proportion des paramètres vitaux observés simultanément dans l'ensemble des visites.

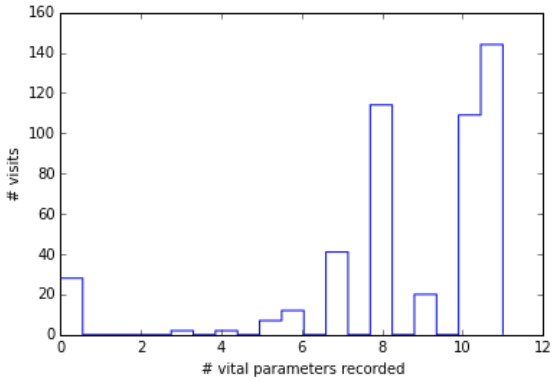

FIGURE 11: Nombre de points par paramètre vital

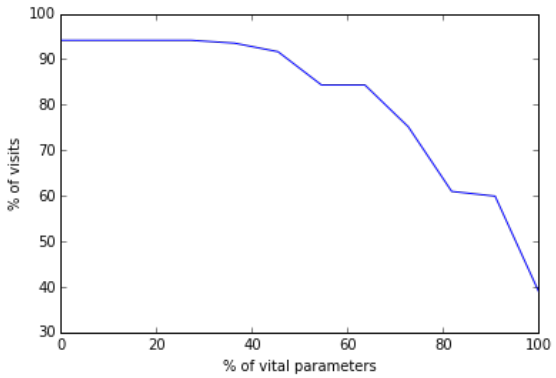

FIGURE 12: Variables biologiques en commun à tous les séjours

Figure	Moyenne	Ecart type	Min	25%	50%	75%	Max
10	8.7	2.7	0.0	8.0	10.0	11.0	11.0

TABLEAU 3 : Statistiques élémentaires pour la figure 10

Ainsi, les paramètres vitaux peuvent quant à eux bien être considérés comme des données longitudinales puisque nous disposons d'un nombre de points suffisant en moyenne pour inférer une cinétique au cours des séjours. De la même façon que pour les variables biologiques, en ne gardant que les paramètres vitaux présents pour au moins 75% des séjours, on parvient aux 9 concepts cités dans la section "présentation des données" et on écarte quelques paramètres sans grande importance a priori, comme la taille des patients. Notons que pour ce qui est du poids, trop peu de points étaient présents par séjour et seules la moyenne du poids du séjour ainsi que la moyenne du gradient ont été prises comme features pour ce paramètre vital.

Le problème principal des paramètres vitaux réside maintenant dans le fait que ces données longitudinales proviennent de séjours à durée variable d'une part, et d'autre part, même si la durée est la même pour deux séjours, les instants d'enregistrements ne sont pas les mêmes ce qui donne lieu à des problèmes d'alignements. Par exemple, on peut observer sur la figure suivante l'évolution de la température pour 5 séjours pris aléatoirement parmi les séjours de patients dits "rechuteurs" (relapse) dans le cadre d'apprentissage supervisé, et 5 séjours pris aléatoirement parmi les autres séjours. Entre parenthèse est précisé le nombre de jours avant la crise suivante, avec "none" si aucun séjour ne suit.

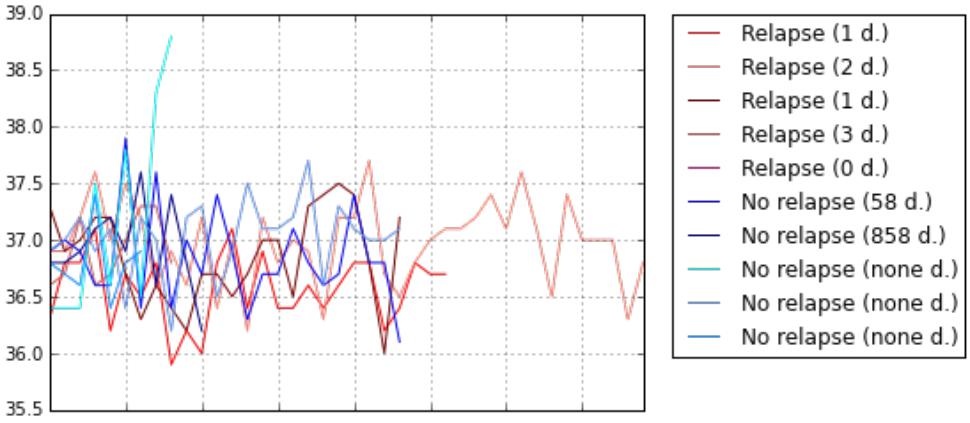

FIGURE 13: Evolution de la température au cours de 10 séjours

Diverses études ont déjà été menées pour faire face à ces problèmes d'alignement [7]. Dans un premier temps et dans le cadre de ce stage, une modélisation relativement simple a été utilisée. Une représentation plus fine de ce type d'information sera par ailleurs au cœur du sujet de la thèse à suivre.

Mais avant d'expliciter la modélisation choisie, précisons d'abord le choix qui a été fait concernant la sélection des séjours. En effet, il s'agit là d'un point important : prendre tous les séjours à notre disposition pour construire notre ensemble d'apprentissage serait une erreur, dans le sens où un certain biais serait probablement introduit par les patients ayant un grand nombre de séjour. Leurs caractéristiques propres seraient alors affectées d'un poids très important, mais à cause de leur nombre de visites, et pas nécessairement parce que celles-ci seraient discriminantes pour la classe à laquelle ces patients appartiennent.
Pour cette raison, un seul séjour sera choisi par patient, ce qui fera un total de 218 séjours. Mais en prenant pour chaque patient un séjour aléatoire, le taux de réhospitalisation précoce (ou rechute) pour les séjours sélectionnés est en moyenne inférieur à 10%, ce qui fait peu de séjours sur un échantillon de 218. Pour augmenter le nombre d'exemples positifs dans l'ensemble d'apprentissage en construction, et pour améliorer le pouvoir de prédiction des algorithmes qui apprendront de celui-ci, la méthode de sélection suivante sera appliquée. Un choix aléatoire sera fait parmi les séjours précédant une rechute pour les patients ayant au moins une rechute, et une sélection aléatoire sera faite sur l'ensemble des séjours des patients n'ayant aucune rechute. On obtient alors un taux de rechute de 12,84%, avec 28 examples positifs sur 218.

Revenons alors sur la stratégie adoptée concernant les données longitudinales et les problèmes d'alignement. Cette dernière est finalement relativement simple : on commence par faire un "rescaling" de la durée de tous les séjours pour se ramener sur le segment $[0, 1]$; avec 0 pour le début et 1 pour la fin du séjour, précédant la sortie et la prédiction. Puis on interpole les données brutes à l'aide de splines du premier ordre (donnant les meilleurs résultats visuellement). Ensuite, on extraira p points espacés d'un intervalle régulier de façon à discrétiser l'information et créer nos features pour les paramètres vitaux. Nous verrons par la suite que nous utiliserons une méthode adaptative pour le choix de p suivant le paramètre vital, mais fixons pour le moment $p = 40$. En plus d'extraire la valeur des concepts suivis, on veut également prendre en compte la cinétique des courbes et extraire p' valeurs de gradient. On utilisera la même méthode adaptative pour le choix de p' qu'on fixe également à 40 pour le moment. Mais avant de calculer le gradient, on applique un filtre de Savitzky–Golay à la série obtenue après interpolation et extraction des p points, de façon à lisser les courbes et débruiter leur évolution.
Avant d'expliciter davantage les détails techniques de la méthode, visualisons le rendu de l'interpolation suivi du filtrage sur quelques échantillons choisis ici pour la fréquence respiratoire. On visualisera en fait les

courbes sans axes car seule l'allule nous intéresse ici. On aura en vert les points réels, puis à gauche et en rouge la courbe interpolée pour un patient positif (donc étant réhospitalisé dans les 15 jours après ce séjour), en rose le résultat de la courbe rouge après application du filtre de Savitzky–Golay ; puis la même chose à droite pour un patient négatif avec le bleu et le cyan remplaçant le rouge et le rose respectivement.

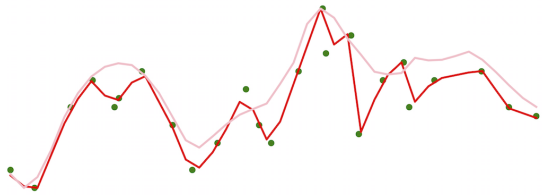

FIGURE 14: Fréquence respiratoire pour un patient positif

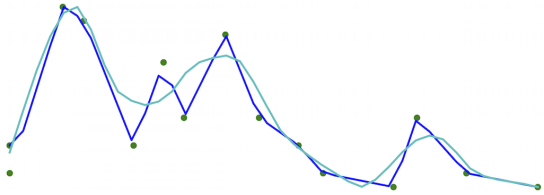

FIGURE 15: Fréquence respiratoire pour un patient négatif

Revenons maintenant rapidement sur les détails techniques de cette construction. Tout d'abord, on utilise la transformation classique suivante pour ramener l'ensemble des durées de séjours au segment $[0, 1]$:

$$\forall j \in [\![1, N]\!],\ x_j = \frac{x_j - min_k(x_k)}{max_k(x_k) - min_k(x_k)}$$

dans le cas où on observe N points.

Après l'étape de rescaling vient celle de l'interpolation. Si on considère que les points réels observés pour un séjour i sont des images issues d'une certaine fonction inconue f_i, régissant par exemple l'évolution de la température d'un patient au cours d'un séjour, l'interpolation de ces points consiste à trouver une approximation de f, notons la $\widetilde{f}_i$, parmi une certaine classe de fonctions, par exemple polynômiale, de façon à ce que les deux fonctions coïncident sur l'ensemble des points, *i.e.* $\forall j \in [\![1, N]\!],\ f_i(x_j) = \widetilde{f}_i(x_j)$ si on observe N points.
Dans notre cas unidimensionnel, les splines sont des fonctions polynômiales par morceaux associées à une partition de l'intervalle sur lequel on observe les points. Celui-ci est ainsi divisé en sous-intervalles plus petits sur lesquels on procède à une interpolation polynômiale classique.
Nous avons utilisé des splines du premier ordre car ceux-ci semblaient donner les meilleures approximations.

Pour tous les séjours i, on prend ensuite l'image par $\widetilde{f}_i$ d'un ensemble de p points uniformement répartis sur le segment $[0, 1]$. Cela donnera les réalisations des p features pour le paramètre vital considéré et pour le séjour i.

Reste alors à prendre l'image par $\nabla\widetilde{f}_i$ de p' points uniformement répartis sur le segment $[0, 1]$, après avoir appliqué un filtre de Savitzky–Golay à $\widetilde{f}_i$ de façon à lisser la courbe avant de calculer les gradients. Le principe du filtre, dont les paramétres ont été choisis après de nombreux tests, est le suivant : une fenêtre glissante dont la longueur a été prise à 9 points est centrée en chacun des points, puis un polynôme de degré 3 est calculé pour approximer au mieux ces 9 points. La fenêtre glissante est progressivement décalée pour visiter tous les points initiaux en son centre. Les points en sortie du filtre sont construits à chaque décalage en prenant l'image par le polynôme d'approximation du centre de la fenêtre.

Si on représente sur un même graphe l'ensemble des courbes en distinguant les deux classes de patients, on ne distingue pas visuellement de différence nette et significative entre les deux classes. Par exemple pour la pression artérielle diastolique (PA min), on a représenté ci-dessous l'ensemble des courbes interpolées, toujours avec $p = 40$, et avec la classe des patients positifs représentée par les nuances de rouge, et les patients négatifs par les nuances de bleu.

En revanche, lorsque l'on moyenne les courbes suivant les deux classes (courbes plus foncées), on voit apparaître des comportements différents. Et le même phénomène est récurent pour l'ensemble des paramètres vitaux, ce qui laisse supposer qu'il y a bien de l'information concernant l'état caché de la crise dans ces courbes.

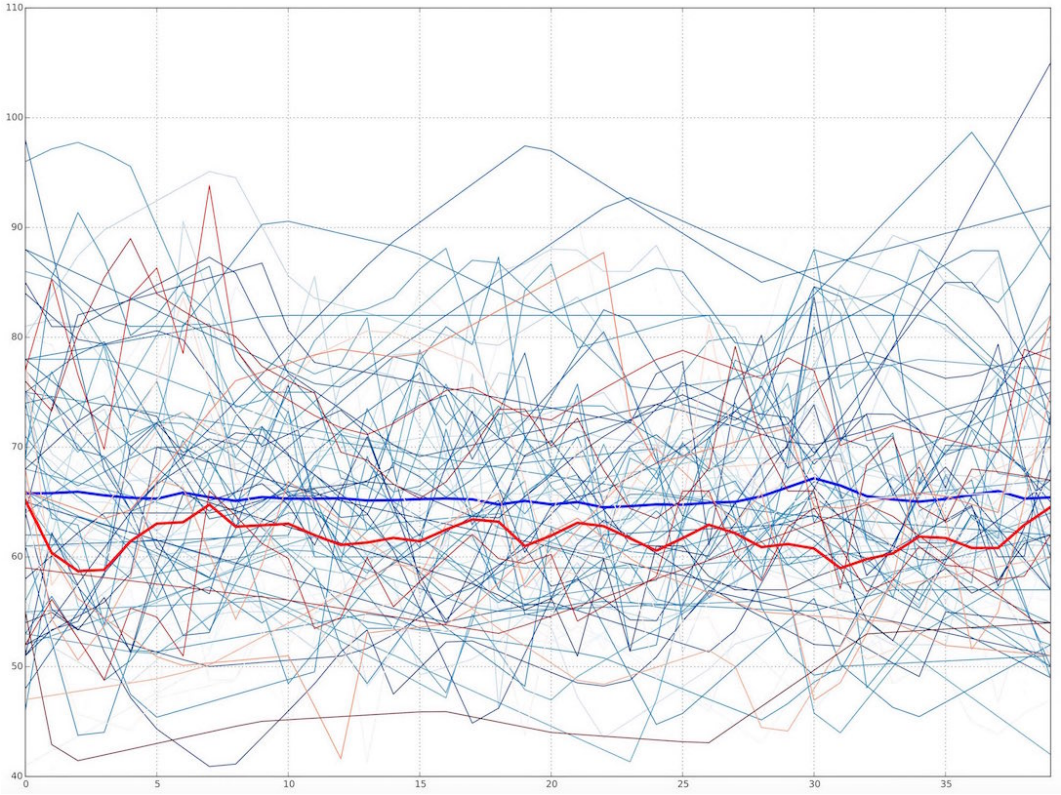

FIGURE 16: Pression artérielle diastolique et moyennes par classe

Enfin, en faisant la même série de tests de Mann-Whitney-Wilcoxon sur l'ensemble des données obtenues concernant les paramètres vitaux que celle réalisée sur les paramètres biologiques, on obtient le type de graphe suivant qui représente uniquement les résultats pour la température. On trouvera sur la figure 48 en annexe l'ensemble des résultats.

On constate que pour les paramètres vitaux également et avec cette modélisation, de l'information réside dans ces données suivant l'avancement du séjour quant à la réhospitalisation précoce.

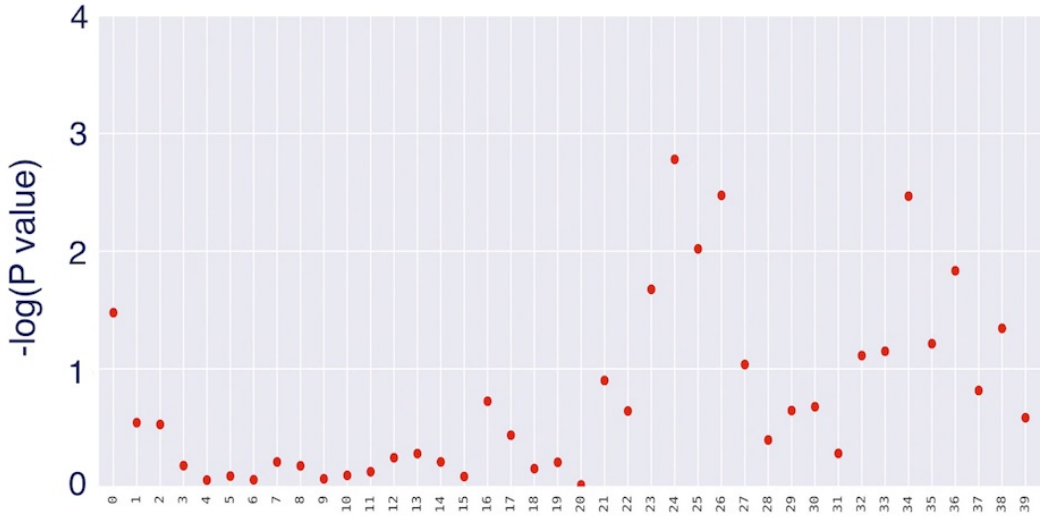

FIGURE 17: Test de Mann-Whitney-Wilcoxon pour la température

Afin de s'assurer qu'il ne s'agit pas de l'effet de quelques points aberrants, une visualisation sous forme de boîtes à moustache des distributions des features crées conditionnellement à la classe binaire a été faite pour les principales variables significatives du test. Des vérifications similaires ont été faites pour les variables biologiques.
Par exemple si on compare les boîtes à moustache conditionnelles du gradient de la pression systolique au niveau du point 37 (donc à la toute fin du séjour), on obtient le graphe suivant :

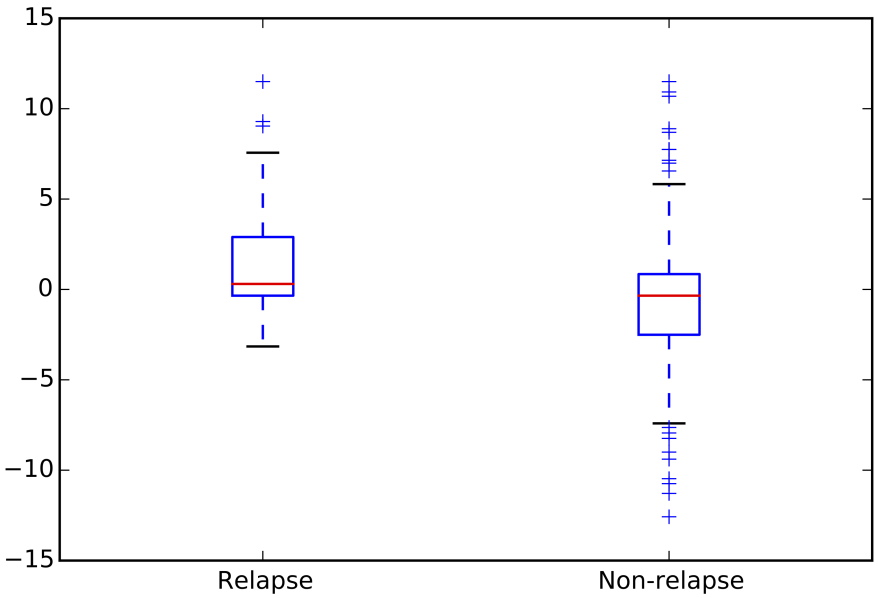

FIGURE 18: Boîtes à moustache du gradient 37 pour la PA max conditionnellement à la classe

Bien que la différence ne soit pas flagrante, il semble en effet que la cinétique pour la pression artérielle en fin de séjour a tendance à augmenter davantage pour les patients à fort risque de rechute.

4.3. Preprocessing et sélection de features

Il existe deux objectifs fondamentaux en apprentissage statistique : assurer des prédictions les meilleures possible, et exhiber les variables prédictives pertinentes. La sélection de variables est particulièrement importante lorsque le vrai modèle sous-jacent posséde une géométrie ou représentation sparse. Identifier les variables prédictives significatives va alors améliorer les performances en prédiction du modèle ajusté [8].

La construction de l'espace de travail va se baser sur ce qui a été décrit dans la section précédente, mais en y incorporant différentes méthodes de préprocessing et de sélection de features, dont le but est notamment d'aboutir à des données bien conditionnées, à faible corrélation, et informatives pour la tâche prédictive visée.

Si on observe la matrice de corrélation linéaire sur l'ensemble des données en gardant p=p'=40 (figure 49 en annexe), on constate une forte corrélation positive pour l'ensemble des valeurs prises au cours des séjours pour les différents paramètres vitaux, ce qui se comprend bien. L'idée est alors de sélectionner les valeurs de p et p' de façon adaptative pour chaque paramètre vital.
On va ainsi construire, pour chaque paramètre vital, l'ensemble $\mathcal{S}$ de features en partant de deux features, une au début du séjour (en 0) et l'autre à la fin (en 1), puis on itère les opérations suivantes :

1. On prend une feature F suplémentaire de façon à ce que l'ensemble des instants sous-jacents soit réparti uniformément sur le segment $[0, 1]$.
2. On calcule la matrice de corrélation linéaire avec l'ensemble des features de $\{\mathcal{S}, F\}$.
3. S'il existe un coefficient de la matrice supérieur en valeur absolue à un certain seuil ε et en dehors de sa diagonale, alors $\mathcal{S}$ est complet et on a en fait $p = Card(\mathcal{S})$. Sinon, $\mathcal{S} = \{\mathcal{S}, F\}$ et on retourne à l'étape 1.

On utilise la même méthode pour sélectionner p et p' et en pratique, on prendra $\varepsilon = 0,95$.
Notons que la matrice de corrélation des variables biologiques seules est représentée figure 50 en annexe et que certaines variables sont très corrélées également. Ceci justifiera une étape de décorrélation sur l'ensemble des données vers la fin du préprocessing.

Rappelons que la matrice de corrélation linéaire est en fait composée des coefficients $\hat{c}_{i,j}$ étant des estimateurs empiriques non biaisés des coefficients de corrélation des features, donnés par $\hat{c}_{i,j} = \dfrac{\hat{\sigma}_{X_i X_j}}{\hat{\sigma}_{X_i}\hat{\sigma}_{X_j}}$,

avec $\hat{\sigma}_{X_i X_j} = \frac{1}{N}\sum_{k=1}^{n}(X_{i,k} - \bar{X}_i)\cdot(X_{j,k} - \bar{X}_j)$, $\hat{\sigma}_{X_i} = \sqrt{\dfrac{1}{N}\sum_{k=1}^{n}(X_{i,k} - \bar{X}_i)^2}$, $\bar{X}_i = \dfrac{1}{N}\sum_{k=1}^{N} X_{i,k}$
qui sont respectivement des estimateurs de la covariance, des écarts-type et des espérances de la variable X_i et de même pour la variable X_j.

Etant donné l'aléa introduit dans la sélection du séjour unique par patient évoquée précédemment, on comprend bien qu'en itérant un grand nombre de fois cette sélection, on va consulter l'ensemble des 479 séjours disponibles. Afin d'améliorer la fiabilité des hyper-paramètres des différentes étapes de selection de features, chacune d'entre elle sera effectuée 50 fois avec l'initialisation aléatoire des choix de séjours, et on prendra par exemple la médiane du paramètre concerné pour la sélection finale. En effet, lorsqu'il s'agira de comparer différents algorithmes d'apprentissage statistique par exemple, on souhaitera que l'espace de travail soit rigoureusement le même et on voudra pouvoir générer les données de façon reproductible, sans risquer de travailler dans des espaces différents.

Pour ce qui est de p et de p' pour les différents paramètres vitaux, on obtient alors les valeurs résumées dans le tableau suivant, où est également précisé l'écart type après les 50 itérations.

Concept	Nombre de points	Ecart type
Fq cardiaque	28	0,3
∇ Fq cardiaque	26	0
PA max	28	0
∇ PA max	26	0
Température	30	8,4
∇ Température	18	0
Saturation O_2	34	4,8
∇ Saturation O_2	26	0
Douleur EVA	21	0,7
∇ Douleur EVA	26	0
Débit O_2	16	4,6
∇ Débit O_2	26	0
Fq respiratoire	21	5,3
∇ Fq respiratoire	26	0
PA min	28	0
∇ PA min	26	0

TABLEAU 4 : Valeurs de p et p' optimales par concept

On remarque donc que les choix de p et p' sont relativement stables. On fixera dès lors leurs valeurs à celles du tableau ci-dessus.

Le nombre de features avec ces paramètres finaux et à ce stade de la phase de sélection est de 446.

Avant de passer à la suite, précisons que les données manquantes sont remplacées par la moyenne des réalisations par features, indépendamment des classes, à ce moment précis du préprocessing.

L'étape suivante va consister à faire un test de Mann-Whitney-Wilcoxon conditionnellement à la classe du séjour, et ce pour chaque feature. Nous allons donc obtenir K=446 *p-values*. L'idée est alors d'appliquer la procédure de Benjamini-Hochberg afin de sélectionner les features les plus informatives pour le futur objectif de prédiction.

Le but de cette méthode est initialement de contrôler le taux de fausse découverte (FDR, pour False Discovery Rate) qui est une quantité permettant de conceptualiser le taux d'erreur de première espèce, c'est-à-dire d'erreur de rejet, lors de multiples tests d'hypothèses supposés indépendants. Contrôler le FDR permet alors de contrôler en moyenne la proportion de rejets incorrects de l'hypothèse nulle $\mathcal{H}_0$, qu'on appelle alors "fausse découverte", comme précisé dans la définition plus formelle qui suit.

Si on a

	On accepte $\mathcal{H}_0$	On rejette $\mathcal{H}_0$
$\mathcal{H}_0$ est vraie	VN (pas d'erreur)	FP (erreur de $1^{ére}$ espèce)
$\mathcal{H}_0$ est fausse	FN (erreur de $2^{éme}$ espèce),	VP (pas d'erreur)

TABLEAU 5 : Table de contingence pour le test de $\mathcal{H}_0$

avec "F" pour "Faux", "V" pour "Vrai", "P" pour "Positif", et "N" pour "Négatif" ; alors

$$FDR = \mathbb{E}\Big[\frac{FP}{FP+VP}\mathbb{1}_{\{FP+VP>0\}}\Big]$$

En pratique, la méthode consiste à suivre la procédure suivante. Les K *p-values* sont tout d'abord triées et rangées par ordre croissant, renommons les :

$$p(1) \leq p(2) \leq \cdots \leq p(K)$$

On détermine ensuite le nombre de features à conserver de cette façon :

$$k = argmax\Big\{j \ : \ p(j) \leq \frac{\alpha j}{K}\Big\}$$

Ce qui correspond en fait à rejeter les hypothèses $\mathcal{H}_0^{(i)}$ correspondant aux *p-values* $p(i)$ pour i dans $[\![1, k]\!]$. Benjamini et Hochberg ont alors montré que cette procédure permet de contrôler le FDR au niveau α.

L'idée étant de faire ici une sélection assez large, nous prendrons en pratique une valeur de α de 0,9 et comme pour les sélections des différents p et p', nous allons itérer la sélection de features 50 fois et conserver les features sélectionnées au moins dans $\frac{1}{3}$ des cas (choix arbitraire). Notons également qu'ici nos tests ne peuvent pas être considérés comme indépendants, et donc on ne peut pas rigoureusement assimiler α à la valeur du FDR. Les résultats à suivre ne sont ainsi pas surprennants, même avec $\alpha = 0,9$ (avec un FDR de 0,9 on se serait attendu à beaucoup plus de variables sélectionnées).
Les features sélectionnées à ce stade sont alors au nombre de 45 et sont résumées par la figure suivante.

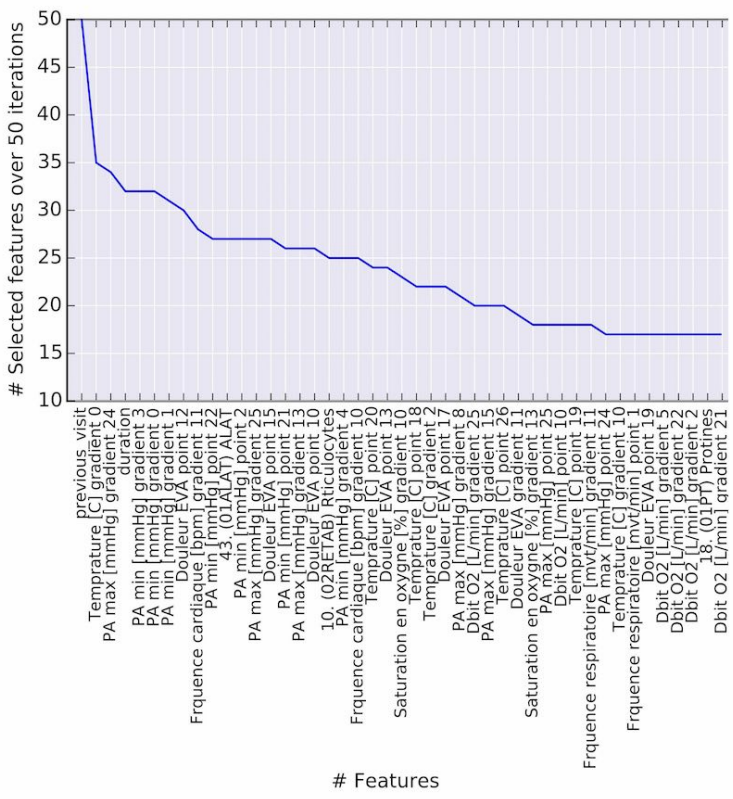

FIGURE 19: Sélection de features par la procédure de Benjamini-Hochberg

La dimension de l'espace à ce stade est donc de 45. On appellera l'étape suivante la binarisation [3]. Comme l'évoque son nom, cette étape va consister à modifier l'espace de travail de façon à n'avoir plus que des features binaires.
Le procédé que nous avons suivi est simple : partant des 45 features quantitatives F_i obtenues à la fin de l'étape précédente, on calcule pour chacune d'entres elles les quartiles associés (indépendamment de la classe) puis on remplace chaque feature quantitative F_i par 4 features binaires $(F_i^1, F_i^2, F_i^3, F_i^4)$, où le séjour j est renseigné à $(0,1,0,0)$ si la $j^{éme}$ réalisation de F_i appartient au deuxième quartile de l'ensemble des observations de F_i. On aboutit alors à un espace de 156 ($\neq 4 \times 45$ car on laisse inchangées les features qui étaient déjà binaires comme le sexe).

Remarque 2 : Une méthode plus évoluée serait de faire ce qu'on appelle un screening TV où on ne prend plus nécessairement les seuils au niveau des quartiles, mais le nombre de variables à créer à partir d'une variable qualitative, ainsi que la position des seuils, deviennent des paramètres appris à l'aide d'une pénalisation par variation totale.
Plus concrétement, pour chaque variable quantitative k dans $[\![1, 45]\!]$, on calcule par exemple les déciles (on peut en pratique être encore plus fin et prendre davantage de quantiles). Puis pour chaque patient i on remplace la feature k par 10 features binaires, qu'on notera $D_i^k \in \{0,1\}^{10}$, composées de 0 et d'un unique 1 au niveau du décile correspondant au patient. On estime alors le vecteur de poids de chacun de ces quantiles de la façon suivante :

$$\hat{\beta}_{TV}^k \in \underset{\beta \in \mathbb{R}^{10}}{argmin} \Big\{ -\sum_{i=1}^{n} Y_i \log(g((D_i^k)^\top \beta)) + (1 - Y_i) \log(1 - g((D_i^k)^\top \beta)) + \lambda \sum_{j=2}^{10} |\beta_j - \beta_{j-1}| \Big\}$$

où $g : x \mapsto \frac{1}{1+e^{-x}}$ est la fonction logistique.
Celà rend ces vecteurs de coefficients longitudinaux constants par morceaux, avec des seuils pour ces morceaux appris automatiquement (position et nombre, le nombre étant contrôlé par l'hyper-paramètre λ choisi par cross-validation). Visuellement, on comprend bien l'intérêt à l'aide de la figure suivante où on a en bleu les coefficients du vecteur de poids (et donc d'importance) qu'on peut obtenir sans cette procédure de screening TV, ce qui peut donc conduire à des résultats peu concluants du point de vue de l'interprétation clinique (par exemple au niveau du deuxième coefficient) ; et en rouge les coefficients appris par cette méthode, où on voit apparaître 2 seuils plutôt que 9.

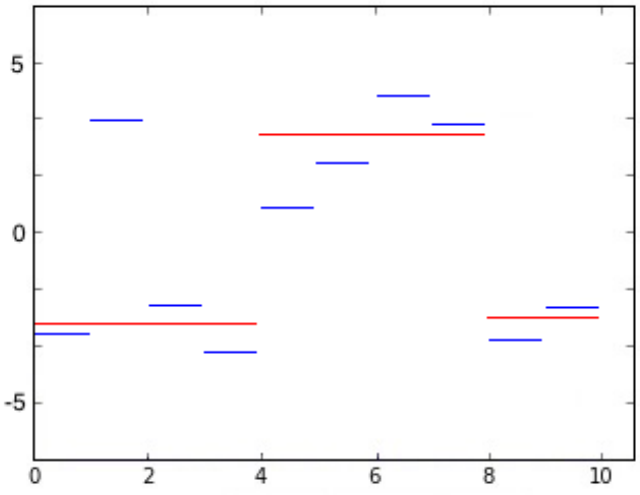

FIGURE 20: Apprentissage automatique des seuils par la méthode du screening TV

Cette méthode sera également testée dans un futur proche.

L'étape qui vient ensuite consiste à ajouter les features d'association, c'est-à-dire l'ensemble des couples (à coordonnées distinctes) du produit cartésien de l'ensemble de features en sortie de l'étape précédente. La dimension de l'espace augmente alors à 11978 ($\simeq \binom{156}{2}$ où on a retiré quelques features, comme celles étant constantes).

Ces features sont importantes à prendre en compte pour des données biologiques, et plus généralement pour des applications médicales, car certaines variables peuvent être peu informatives prises séparément, mais avoir un pouvoir prédictif important si elles sont simultanément dans certaines régions de l'espace.

Comme évoqué au début de cette section, une étape de décorrélation à part entière de l'ensemble des données reste nécessaire pour travailler ensuite dans de bonnes conditions. C'est pourquoi on calcule à ce stade la matrice de corrélation linéaire sur l'ensemble des features, et si certains coefficients sont supérieurs en valeur absolue à 0.95, alors on conserve aléatoirement l'une des deux features relatives au coefficient.
Le nombre de features rejetées par cette procédure est de 1404, et la dimension de l'espace est de 10574. On peut alors visualiser la matrice de corrélation finale sur la figure 51 en annexe et constater qu'on a bien retiré les trop fortes corrélations.

Remarque 3 : Ayant affaire ici à des features exclusivement binaires, on aurait aussi pu calculer des distances du χ^2 pour traiter la question de l'indépendance des variables plutôt que d'utiliser la corrélation linéaire, les résultats auraient été globalement semblables.

Enfin, on peut visualiser sur le schéma suivant les différentes étapes que nous avons suivi dans cette section.

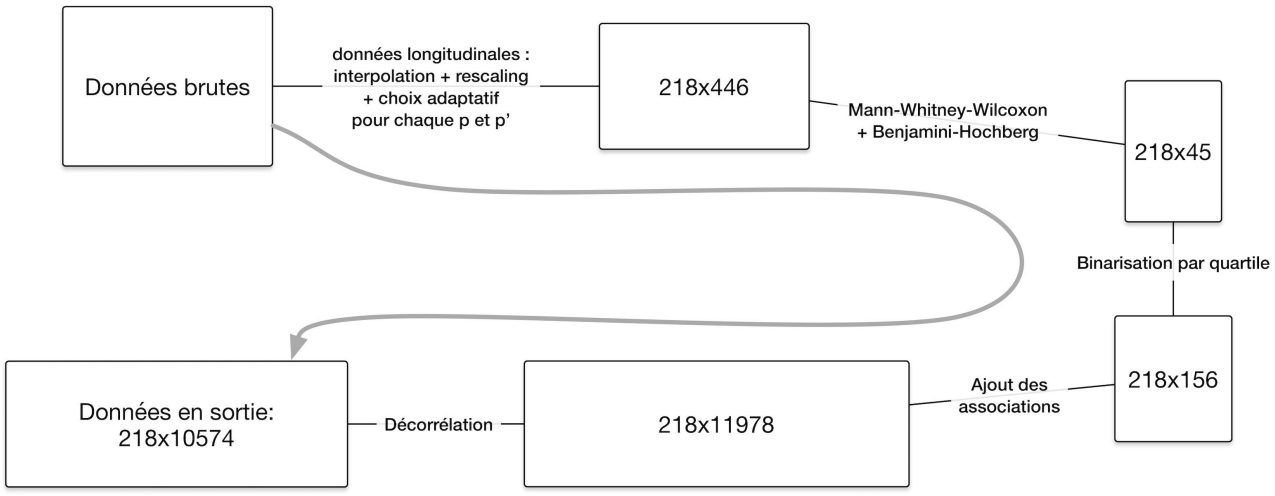

FIGURE 21: Les différentes étapes suivies lors du préprocessing

5. Modèle statistique considéré

Nous allons décrire dans cette section le modèle statistique considéré après l'étude détaillée des données extraites puis nettoyées.

5.1. Un modèle de mélange

Commençons par fixer proprement le cadre de travail. On dispose d'un échantillon d'apprentissage :

$$(X_1, T_1)\ , \ldots, (X_n, T_n) \in \mathbb{R}^d \times \mathbb{N}^*\ i.i.d. \sim \mathbb{P}$$

où $n \in \mathbb{N}^*$ représente le nombre de séjours considérés, d le nombre de variables observées au cours des séjours, $\forall i \in [\![1, n]\!]$, $\begin{cases} X_i \in \mathbb{R}^d\ v.a. \text{ modélisant l'ensemble des données observées pour le séjour } i \\ T_i \in \mathbb{N}^*\ v.a. \text{ modélisant le nombre de jours écoulés entre le séjour } i \text{ et le séjour suivant} \end{cases}$, et enfin $\mathbb{P}$ la mesure de probabilité sous-jacente régissant les phénomènes observés, que l'on cherchera à estimer du mieux possible par une loi $\mathbb{P}_\theta$ avec $\theta \in \Theta$ où Θ sera l'espace des paramètres et $\{\mathbb{P}_\theta, \theta \in \Theta\}$ la famille de lois considérée dans le modèle statistique.

Remarque 4 : On voit dès lors apparaître un premier problème de modélisation concernant la variable aléatoire des temps de retour pour les patients n'ayant qu'un seul séjour à l'hôpital, ou plus généralement pour tous les derniers séjours des patients. Cela induit un phénomène de censure que nous discuterons en détail par la suite, mais commençons dans un premier temps par laisser ce problème de côté et faire comme si toutes les réalisations des T_i étaient bien des valeurs entières ayant un sens réel.

Remarque 5 : On a choisi ici d'exprimer les temps de retour en jours donc par des entiers naturels, mais un modèle continue aurait été similaire.

On cherche alors à exploiter le fait que les patients ayant un risque plus élevé d'être réhospitaliser de façon précoce vont se différencier des patients à plus faible risque par une distribution des temps de retour différente. Nous allons ainsi modéliser un mélange de réponses des distributions des durées inter-visites.

Nous ferons par la suite différentes hypothèses qui tiendront pour les raisonnements futurs.
Hypothèse 1 : On suppose l'existence de $Z \in \{0, 1\}\ v.a.$ latente telle que

$$\forall i \in [\![1, n]\!],\ \forall k \in \{0, 1\},\ T_i | Z_i = k \sim \ell_k$$

où ℓ_k est une loi qui dépend de la réalisation de la variable aléatoire latente, soit de la classe cachée du patient. On dira alors que le patient i a un faible risque de réhospitalisation précoce si $Z_i = 0$ et un fort risque si $Z_i = 1$.
On a alors

$$\forall i \in [\![1, n]\!],\ \forall t \in \mathbb{N}^*, \mathbb{P}(T_i = t | X_i) = \mathbb{P}(T_i = t | Z_i = 0, X_i)\mathbb{P}(Z_i = 0 | X_i) + \mathbb{P}(T_i = t | Z_i = 1, X_i)\mathbb{P}(Z_i = 1 | X_i)$$

Ce qu'on notera

$$\forall i \in [\![1, n]\!],\ \forall t \in \mathbb{N}^*, f_{T_i|X_i}(t) = \pi_0(X_i) f_{T_i|X_i,0}(t) + (1 - \pi_0(X_i)) f_{T_i|X_i,1}(t)$$

avec $\pi_0(X_i)$ la probabilité que le patient i ne donne pas lieu à une réhospitalisation précoce au vu des données X_i observées lors de son séjour. On a donc un modèle de mélange de lois, nous verrons par la suite les différentes lois considérées.

Remarque 6 : Depuis le début de cette section, aucun seuil n'a été évoqué quant au nombre de jours nécessaire pour considérer qu'un retour est précoce ou non (cf. les 15 jours évoqués par les cliniciens). On peut alors

voir la variable cachée comme modélisant le "niveau de risque", ou "l'état de la crise" à la sortie du patient, indiquant si celle-ci est terminée et correctement traitée ou non.

Hypothèse 2 : On suppose que

$$\forall i \in [\![1,n]\!],\ \log \frac{\mathbb{P}(Z_i=1|X_i)}{\mathbb{P}(Z_i=0|X_i)} = b_0 + \sum_{j=1}^{d} b_j X_i^j \tag{1}$$

En posant $\beta = (b_0, \dots, b_d)^\top \in \mathbb{R}^{p+1}$ et en notant indifféremment X_i et $(1, X_i^1, \dots, X_i^d)$, on a alors

$$(1) \Leftrightarrow \pi_0(X_i) = \frac{1}{1+e^{-X_i^\top \beta}} \tag{2}$$

Soit donc $\pi_0(X_i)$ une fonction logistique en les variables observées X_i, et le vecteur de poids β dont chaque coefficient quantifie l'impact de chaque variable observée sur la probabilité du patient i d'appartenir à l'une ou l'autre des deux classes.

Hypothèse 3 : On suppose que $\forall i \in [\![1,n]\!], Z_i \sim \mathcal{B}(1-\pi_0(X_i))$ loi de Bernoulli de paramètre $1-\pi_0(X_i)$.

Les choix que nous considérerons dans la suite pour les lois ℓ_k seront des lois géométriques de paramétre p_k, notées $\mathcal{G}(p_k)$ et de densité $\forall t \in \mathbb{N}^*, f_k(t) = p_k(1-p_k)^{t-1}$, et nous avons aussi fait des essais avec des lois de Pareto discrètes, notées $\mathcal{P}(\alpha_k)$ et de densité $\forall t \in \mathbb{N}^*, f_k(t) = \frac{1}{t^{\alpha_k}} - \frac{1}{(t+1)^{\alpha_k}}$.

Remarque 7 : D'autres essais de lois ont été imaginés et restent à tester, comme la loi de Weibull [1] ou la loi bêta.

5.2. Inférence pour ce modèle

On peut écrire le modèle d'une manière Bayesienne hiérarchique comme suit : on génère d'abord l'état caché $Z_i \in \{0,1\}$ du patient selon une distribution de Bernoulli $\mathcal{B}(1-\pi_0(X_i))$, et on génère ensuite le temps de retour T_i conditionnellement à Z_i selon ℓ_k.

Commençons d'abord par le cas du mélange de deux lois géométriques. Les paramètres du modèle à estimer sont alors $\theta = \{p_0, p_1, \beta\} \in \mathbb{R}^{d+3}$.

Ecrivons la log-vraissemblance du modèle (normalisée par le nombre d'exemples de l'ensemble d'apprentissage) que l'on va chercher à maximiser :

$$\begin{aligned}\ell_n(\theta) &= \frac{1}{n}\log \prod_{i=1}^{n} \mathbb{P}_\theta(T_i|X_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \log\Big[\frac{p_0(1-p_0)^{T_i-1}}{1+e^{-X_i^\top\beta_0}} + \frac{p_1(1-p_1)^{T_i-1}}{1+e^{X_i^\top\beta_0}}\Big]\end{aligned}$$

On aboutit au problème d'optimisation suivant, où on applique une régularisation de type Elastic-Net afin d'éviter le sur-apprentissage et améliorer le pouvoir de prédiction.

$$\hat{\theta} \in \underset{(p_0,p_1,\beta)\in]0,1[^2\times\mathbb{R}^{d+1}}{argmin} \Big\{ -\frac{1}{n}\sum_{i=1}^{n} \log\Big[\frac{p_0(1-p_0)^{T_i-1}}{1+e^{-X_i^\top\beta_0}} + \frac{p_1(1-p_1)^{T_i-1}}{1+e^{X_i^\top\beta_0}}\Big] + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2 \Big\}.$$

Les hyper-paramètres λ_1 et λ_2 pourront être choisis par une K-fold cross-validation, comme nous le verrons par la suite.

Remarque 8 : On pourrait faire dépendre les paramètres p_0 et p_1 des features X_i en ajoutant des coefficients, cela n'affectera pas la suite du raisonnement.

6. Algorithmes d'apprentissage statistique utilisés

6.1. Choix des algorithmes

De nombreux algorithmes ont été essayés dans le cadre de l'apprentissage supervisé, l'idée a alors été de garder les algorithmes donnant les meilleures performances, mais aussi ceux permettant une interprétation clinique exploitable. Par chance, parmi les différents essais, la régression logistique a figuré parmi les algorithmes donnant lieu aux meilleures prédictions, donnant une interprétation clinique très simple à extraire du modèle.

Mais la régression logistique prédit une probabilité de retour dans les 15 jours après la sortie du patient, et n'exploite pas le modèle de mélange développé dans la section précédente. On utilisera alors un algorithme Espérance-Maximisation (EM) [6] pour inférer le θ construit précédemment.

D'autres méthodes d'apprentissage ont été imaginées au cours du stage, mais n'ont pas été pleinement exploitées par manque de temps. Celles-ci seront approfondies aux cours des travaux suivant la fin du stage pour consolider l'article en création concernant les résultats du stage. Citons rapidement deux des méthodes restant à essayer.

Tout d'abord, pour tenter d'améliorer les résultats et l'erreur d'entraînement sans pour autant sur-apprendre, on pourrait appliquer une méhode de boosting en utilisant plusieurs classifieurs de régression logistique. Celà est possible selon la théorie sous-jacente au principe du boosting stipulant que la complexité de Rademacher d'une classe de classifieurs ne change pas si on prend l'enveloppe convexe de cette classe.

$$\overline{\mathcal{R}}_n(\text{conv}(\mathcal{G})) = \overline{\mathcal{R}}_n(\mathcal{G})$$

où la complexité de Rademacher est une façon classique de quantifier la complexité d'une classe de classifieur définie ci-après. Commençons par définir la complexité de Rademacher empirique. Si on note $X^n = \{X_1, \dots, X_n\}$ notre échantillon de points de $\mathbb{R}^d$ et $\mathcal{G}$ la classe de régles de classifications binaires sur $\mathbb{R}^d$ issue de la régression logistique, *i.e.* $\mathcal{G} \subseteq \{g : \mathbb{R}^d \to \{-1, 1\}\}$ avec

$$\forall X \in \mathbb{R}^d, g(X) = \mathbb{1}_{\{\frac{1}{1+e^{-X^\top \beta}} > 0,5\}} - \mathbb{1}_{\{\frac{1}{1+e^{-X^\top \beta}} \leq 0,5\}}$$

où β est un vecteur de poids appris par une méthode de régression logistique régularisée. Soient alors $\varepsilon_1, \dots, \varepsilon_n$ *v.a. i.i.d.* uniformément distribuées sur $-1, 1$, *i.e.* $\mathbb{P}(\varepsilon_i = 1) = \mathbb{P}(\varepsilon_i = -1) = \frac{1}{2}$ qu'on appelle variables aléatoires de Rademacher. Alors la complexité de Rademacher empirique est la variable aléatoire (aléatoire en l'échantillon) $\hat{\mathcal{R}}_n(\mathcal{G}) = \mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\left[\sup_{g\in\mathcal{G}}(\frac{1}{n}\sum_{i=1}^n \varepsilon_i g(X_i))\right]$ (quantifie le "pouvoir" la classe $\mathcal{G}$ à apprendre un bruit aléatoire). Alors la complexité de Rademacher est simplement $\overline{\mathcal{R}}_n(\mathcal{G}) = \mathbb{E}_{X_1,\dots,X_n}\left[\hat{\mathcal{R}}_n(\mathcal{G})\right]$.

Or pour prouver le résultat annoncé, il suffit d'écrire

$$\begin{aligned}\hat{\mathcal{R}}_n(conv(\mathcal{G})) &= \frac{1}{n}\mathbb{E}\Big[\sup_{M\geq 0, g_i\in\mathcal{G}, \alpha_i\geq 0, \sum_{i=1}^M \alpha_i = 1}\Big(\sum_{i=1}^n \varepsilon_i \sum_{j=1}^M \alpha_j g_j(X_i)\Big)\Big] \\ &= \frac{1}{n}\mathbb{E}\Big[\sup_{M\geq 0}\ \sup_{g_i\in\mathcal{G}}\Big[\sup_{\alpha_i\geq 0, \sum_{i=1}^M \alpha_i = 1}\Big(\sum_{i=1}^n \varepsilon_i \sum_{j=1}^M \alpha_j g_j(X_i)\Big)\Big]\Big] \\ &= \hat{\mathcal{R}}_n(\mathcal{G})\end{aligned}$$

Une autre méthode d'apprentissage que nous essaierons pourra être les forêts aléatoires qui prennent bien en compte les interactions entre les variables biologiques et qui sont très utilisées en médecine [11], [4].

Remarque 9 : Les méthodes de boosting ou les forêts aléatoires ont cependant des propriétés d'interprétations cliniques moins intéressantes que la régression logistique, et c'est en partie pourquoi les efforts n'ont pas été concentrés sur ces algorithmes. Ceux-ci seront cependant à tester dans l'optique d'améliorer le pouvoir prédictif.

Enfin, la régression de Cox, et plus généralement les modèles de survie, seront à explorer davantage pour prendre en compte la censure [12].

6.2. L'approche régression logistique

Rappelons tout d'abord le principe général de la régression logistique que nous allons utiliser dans la suite.

La *log*-vraissemblance du modèle s'écrit après quelques transformations classiques de la façon suivante

$$\ell_n(\beta) = \textstyle\sum_{i=1}^n Y_i \log(g(X_i^\top \beta)) + (1 - Y_i)\log(1 - g(X_i^\top \beta)),$$

où g est toujours la fonction logistique.

Si on note $\mathbf{X}$ la matrice de design, $\mathbf{Y}$ le vecteur des n labels binaires et $\mathbf{g} = \Big(g(X_i^\top \beta)\Big)_{i\in[\![1,n]\!]}$, alors le gradient de la *log*-vraissemblance s'écrit $J(\beta) = \mathbf{X}^\top[\mathbf{Y} - \mathbf{g}]$ et la matrice Hessienne $H(\beta) = -\mathbf{X}^\top \mathbf{R}\mathbf{X}$ avec $\mathbf{R}_{i,i} = g(X_i^\top \beta)(1 - g(X_i^\top \beta))$.

La régression logistique consiste alors simplement à itérer l'algorithme de Newton-Raphson $\beta^{(t+1)} = \beta^{(t)} + H(\beta)^{-1}J(\beta)$ jusqu'à un certain seuil de précision sur β (problème d'optimisation convexe donc convergence guarantie).

Remarque 10 : En plus de cette construction probabiliste, on peut aussi voir la régression logistique de façon plus générale comme étant la minimisation du risque empirique convexifié $\hat{A}_n(f) = \frac{1}{n}\sum_{i=1}^n[\varphi(-Y_i f(X_i))]$ sur l'ensemble des régles de décisions $f : \mathbb{R}^d \to \mathbb{R}$ linéaires, correspondant aux classifieurs binaires $g_f : x \mapsto \mathbb{1}_{\{f(x)>0\}} - \mathbb{1}_{\{f(x)\leq 0\}}$; et avec $\varphi : x \mapsto \log(1 + e^x)$ la perte logistique.

Etant donné que nous travaillons dans un espace à grande dimension, le classifieur va avoir beaucoup de liberté, et pour éviter le sur-apprentissage, nous allons ajouter de la régulatisation pour contrôler la complexité et le compromis biais-variance. De cette façon, on cherche à construire un classifieur ayant de bonnes propriétés de généralisation. L'algorithme décrit précédemment s'adapte alors facilement pour respecter les

précisions à suivre.

Pour la régression ridge, on ajoute un terme de régularisation ℓ_2 de façon à pénaliser des poids β_i trop importants. On aura alors

$$\hat{\beta}_{ridge} \in \underset{\beta\in\mathbb{R}^d}{argmin}\Big\{ -\sum_{i=1}^{n} Y_i \log(g(X_i^\top \beta)) + (1-Y_i)\log(1-g(X_i^\top\beta)) + \lambda_2 \|\beta\|_2^2 \Big\}$$

avec $\|\beta\|_2 = \sum_{i=1}^d \beta_i^2$, et où λ_2 sera choisi par cross-validation.

Pour la régression lasso, on ajoutera cette fois un terme de régularisation ℓ_1 de façon à forcer les poids β_i à s'annuler. Cette procédure permet, comme on le verra, de faire une ultime sélection de variables et d'effectuer un second apprentissage sur le support du lasso [15]. On aura

$$\hat{\beta}_{lasso} \in \underset{\beta\in\mathbb{R}^d}{argmin}\Big\{ -\sum_{i=1}^{n} Y_i \log(g(X_i^\top \beta)) + (1-Y_i)\log(1-g(X_i^\top\beta)) + \lambda_1 \|\beta\|_1 \Big\}$$

avec $\|\beta\|_1 = \sum_{i=1}^d |\beta_i|$.

Afin d'évaluer les performances du classifieur résultant de cette approche, nous utiliserons comme métrique le score AUC qui correspond à l'air en dessous de la courbe *ROC*. Rappelons alors que la courbe *ROC* d'une certaire règle de décision réelle $s : \mathbb{R}^d \to \mathbb{R}$ dans un problème de classification binaire dans $\{-1, 1\}$ est la courbe paramétrée suivante.

$$\begin{aligned} ROC : \mathbb{R} &\to [0,1]^2 \\ t &\mapsto \Big(\mathbb{P}(s(X) \geq t | Y = -1), \mathbb{P}(s(X) \geq t | Y = 1)\Big) \end{aligned}$$

où $\mathbb{P}(s(X) \geq t | Y = 1)$ peut être vu comme le taux de vrais positifs du classifieur binaire

$$g_{s,t} : X \mapsto \mathbb{1}_{\{s(X)\geq t\}} - \mathbb{1}_{\{s(X)<t\}},$$

et $\mathbb{P}(s(X) \geq t | Y = -1)$ comme son taux de faux positifs.

Le score *AUC* associé à un classifieur sera alors un réel dans $[\frac{1}{2}, 1]$ correspondant à l'aire sous sa courbe *ROC*, calculé sur un échantillon de test jusqu'alors inobservable. On comprend alors que plus ce score sera proche de 1, meilleures seront les performances du classifieur, et que la courbe *ROC* est par définition insensible aux cardinaux des classes prédites. Le score *AUC* est ainsi tout à fait adapté à notre cadre de travail, étant donné que nous avons affaire à un jeu de données déséquilibré en les proportions des classes des exemples. Mais pour avoir un regard tout à fait complet sur les classifieurs que nous considérerons, nous regarderons également attentivement les scores de précision et de recall, définis ci-après. En reprenant les notations du Tableau 5,

	$Y^* = 1$	$Y^* = -1$
$\hat{Y} = 1$	VP	FP
$\hat{Y} = -1$	FN	VN

TABLEAU 6 : Table de contingence pour la prédiction binaire

Alors la précision et le recall sont définis respectivement par $\dfrac{VP}{VP+FP}$ et $\dfrac{VP}{VP+FN}$.

6.3. L'approche EM

Ecrivons dans cette partie les différentes étapes de l'algorithme EM qui nous permettront d'inférer les paramètres θ explicités plus haut, en commençant par le mélange de deux lois géométriques.

Remarque 11 : Par soucis d'écriture, on omettra dans la suite le conditionnement par rapport aux features $(X_1, \dots, X_n)$, mais on n'oubliera pas que tous les calculs dans ce qui suit sont faits conditionnellement à celles-ci.
En notant $\begin{cases} \mathcal{T} = (T_1, \dots, T_n) \\ \mathcal{Z} = (Z_1, \dots, Z_n) \end{cases}$, on a alors la log-vraissemblance du modèle complété qui s'écrit de la façon suivante (normalisée par le nombre d'exemples) :

$$\begin{aligned}
\ell_n^c(\theta, \mathcal{T}, \mathcal{Z}) &= \frac{1}{n}\sum_{i=1}^n \log \mathbb{P}_\theta(T_i, Z_i) \\
&= \frac{1}{n}\sum_{i=1}^n \log(\mathbb{P}_\theta(Z_i)\mathbb{P}_\theta(T_i|Z_i)) \\
&= \frac{1}{n}\sum_{i=1}^n \log\Big[\pi_0(X_i)^{1-Z_i}(1-\pi_0(X_i))^{Z_i}\Big(p_0(1-p_0)^{T_i-1}\Big)^{1-Z_i}\Big(p_1(1-p_1)^{T_i-1}\Big)^{Z_i}\Big] \\
&= \frac{1}{n}\sum_{i=1}^n \Big[Z_i\Big(\log(1-\pi_0(X_i)) + \log p_1 + (T_i-1)\log(1-p_1)\Big) \\
&\qquad + (1-Z_i)\Big(\log(\pi_0(X_i)) + \log p_0 + (T_i-1)\log(1-p_0)\Big)\Big]
\end{aligned}$$

Décrivons alors l'iteration $l+1$ de l'algorithme.

Etape E : Notons $\mathcal{Q}_n(\theta, \theta^{(l)}) = \mathbb{E}_{\theta^{(l)}}[\ell_n^c(\theta, \mathcal{T}, \mathcal{Z})|\mathcal{T}]$ que l'on cherche à calculer à l'iteration $l+1$. Cela revient en fait à remplacer dans l'expression précédente les Z_i par $q_i^{(l)} = \mathbb{E}_{\theta^{(l)}}[Z_i|T_i]$, étant donné que tous les autres termes sont constants du point de vue de la probabilité conditionnelle de Z sachant T.

Remarque 12 : On peut écrire $q_i^{(l)}$ en "oubliant" la dépendance en $\theta^{(l)}$, étant donné que $\theta^{(l)}$ sera fixé lors de l'étape de Maximisation suivante.

On a alors pour tout i dans $[\![1, n]\!]$,

$$\begin{aligned}
q_i^{(l)} &= \mathbb{E}_{\theta^{(l)}}[\mathbb{1}_{\{Z_i=1\}}|T_i] \\
&= \mathbb{P}_{\theta^{(l)}}[Z_i = 1|T_i] \\
&= \frac{\mathbb{P}_{\theta^{(l)}}[T_i|Z_i=1]\mathbb{P}_{\theta^{(l)}}[Z_i=1]}{\mathbb{P}_{\theta^{(l)}}[T_i]} \\
&= \frac{p_1^{(l)}(1-p_1^{(l)})^{T_i-1}(1-\pi_0(X_i))}{p_0^{(l)}(1-p_0^{(l)})^{T_i-1}\pi_0(X_i) + p_1^{(l)}(1-p_1^{(l)})^{T_i-1}(1-\pi_0(X_i))}
\end{aligned}$$

Et donc

$$\begin{aligned}
\mathcal{Q}_n(\theta, \theta^{(l)}) = \frac{1}{n}\sum_{i=1}^n \Big[&q_i^{(l)}\Big(\log(1-\pi_0(X_i)) + \log p_1 + (T_i-1)\log(1-p_1)\Big) \\
&+ (1-q_i^{(l)})\Big(\log(\pi_0(X_i)) + \log p_0 + (T_i-1)\log(1-p_0)\Big)\Big]
\end{aligned}$$

Etape M : On cherche maintenant à maximiser la quantité précédente par rapport à θ. La mise à jour de p_0 et p_1 est explicite, il suffit d'annuler le gradient calculé par rapport à ces deux variables respectivement.

$$\frac{\partial \mathcal{Q}_n(\theta, \theta^{(l)})}{\partial p_0} = \sum_{i=1}^{n} (1 - q_i^{(l)}) \frac{1}{p_0} - \frac{(T_i - 1)(1 - q_i^{(l)})}{1 - p_0}$$

Et donc

$$\begin{aligned} \frac{\partial \mathcal{Q}_n(\theta, \theta^{(l)})}{\partial p_0} = 0 \quad &\Leftrightarrow \quad p_0^{(l+1)} \Big(\sum_{i=1}^{n} (T_i - 1)(1 - q_i^{(l)}) + \sum_{i=1}^{n} (1 - q_i^{(l)}) \Big) = \sum_{i=1}^{n} (1 - q_i^{(l)}) \\ &\Leftrightarrow \quad p_0^{(l+1)} = \frac{n - \sum_{i=1}^{n} q_i^{(l)}}{\sum_{i=1}^{n} T_i (1 - q_i^{(l)})} \end{aligned}$$

De même

$$\frac{\partial \mathcal{Q}_n(\theta, \theta^{(l)})}{\partial p_1} = \sum_{i=1}^{n} \frac{q_i^{(l)}}{p_1} - \frac{(T_i - 1) q_i^{(l)}}{1 - p_1}$$

Et

$$\begin{aligned} \frac{\partial \mathcal{Q}_n(\theta, \theta^{(l)})}{\partial p_1} = 0 \quad &\Leftrightarrow \quad p_1^{(l+1)} \sum_{i=1}^{n} (T_i - 1) q_i^{(l)} = (1 - p_1^{(l+1)}) \sum_{i=1}^{n} q_i^{(l)} \\ &\Leftrightarrow \quad p_1^{(l+1)} = \frac{\sum_{i=1}^{n} q_i^{(l)}}{\sum_{i=1}^{n} T_i q_i^{(l)}} \end{aligned}$$

La mise à jour de β n'est pas explicite et requiert quant à elle quelques itérations d'un algorithme d'optimisation non linéaire sans contrainte. On utilisera pour cela une méthode quasi-Newton avec l'algortihme L-BGFS-B [16, 10], qui nécessite le gradient par rapport à β et qui utilise une approximation de l'inverse de la matrice Hessienne.

Or, en notant $-R_n^{(l)}(\beta)$ la quantité de $\mathcal{Q}_n(\theta, \theta^{(l)})$ qui dépend de β, on remarque que

$$\begin{aligned} -R_n^{(l)}(\beta) &= \frac{1}{n} \sum_{i=1}^{n} \Big[q_i^{(l)} \log(1 - \pi_0(X_i)) + (1 - q_i^{(l)}) \log(\pi_0(X_i)) \Big] \\ &= \frac{1}{n} \sum_{i=1}^{n} \Big[q_i^{(l)} \log \Big(\frac{1 - \pi_0(X_i)}{\pi_0(X_i)} \Big) + \log(\pi_0(X_i)) \Big] \\ &= \frac{1}{n} \sum_{i=1}^{n} - \Big[q_i^{(l)} X_i^\top \beta + \log(1 + e^{-X_i^\top \beta}) \Big] \end{aligned}$$

qui vient du fait que par construction et grâce à la fonction logistique, le log odd-ratio est linéaire.
On obtient alors

$$\begin{aligned}\frac{\partial \mathcal{Q}_n(\theta,\theta^{(l)})}{\partial \beta} &= \frac{\partial R_n^{(l)}(\beta)}{\partial \beta} \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big[q_i^{(l)} - \frac{1}{1+e^{-X_i^\top \beta}}\Big]X_i \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big[q_i^{(l)} - \pi_0(X_i)\Big]X_i\end{aligned}$$

La mise à jour de β requiert également la prise en compte des termes de régularisation et on cherche en fait à minimiser la quantité

$$-\mathcal{Q}_n(\theta,\theta^{(l)}) + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2\,; \tag{3}$$

ce qui revient donc à minimiser

$$R_n^{(l)}(\beta) + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2. \tag{4}$$

La minimisation du problème (4) est un problème convexe qui est donc effectuée par l'algorithme L-BGFS-B. Pour utiliser cet algorithme avec la pénalisation ℓ_1, qui n'est pas différentiable, on utilisera l'astuce suivante. Pour tout réel $a \in \mathbb{R}$, on a $|a| = a_+ + a_-$ avec a_+ et a_- qui sont respectivement la partie positive et la partie négative de a, on impose alors les contraintes $a_+ \geq 0$ et $a_- \geq 0$. On ré-écrit le problème de minimisation (4) sous la forme du problème différentiable sous contrainte suivant

$$\begin{aligned}\text{minimiser} \quad & R_n^{(l)}(\beta^+ - \beta^-) + \lambda_1\sum_{j=1}^{d}\beta_j^+ + \lambda_1\sum_{j=1}^{d}\beta_j^- + \frac{\lambda_2}{2}\|\beta^+ - \beta^-\|_2^2 \\ \text{tel que} \quad & \forall j \in [\![1,d]\!], \beta_j+ \geq 0 \text{ et } \beta_j^- \geq 0\end{aligned} \tag{5}$$

L'Algorithme 1 décrit alors les étapes de l'algorithme EM pour minimiser le problème (5).

Algorithm 1 L'algorithme EM pour l'inférence du modèle de mélange

Require: Les données $(X_i, T_i)_{i\in[\![1,n]\!]}$; paramètres initiaux $p_0^{(0)}, p_1^{(0)}, \beta^{(0)}$; hyper-paramètres $\lambda_1, \lambda_2 \geq 0$

1: **for** $l = 0, \ldots,$ until convergence **do**
2: Calculer $q_i^{(l)}$
3: Calculer $p_0^{(l+1)}$ et $p_1^{(l+1)}$
4: Calculer $\beta^{(l+1)}$ en résolvant (5) avec l'algorithme L-BGFS-B
5: **end for**
6: **return** $p_0^{(l)}, p_1^{(l)}$ and $\beta^{(l)}$

Pour initialiser les paramètres, on peut utiliser le vecteur nul pour β et calculer $p_0^{(0)}$ et $p_1^{(0)}$ à l'aide des estimateurs du maximum de vraissemblance d'une distribution mélange de lois géométriques sur les labels observés $(T_i)_{i\in[\![1,n]\!]}$, comme nous le verrons dans la section 8.2. concernant les tests sur les données réelles.

Remarque 13 : Afin d'éviter les problèmes d'overflow, notons qu'en pratique, on s'assurera de calculer uniquement des exponentiels de nombre négatifs, en codant par exemple $\frac{1}{1+e^{-x}}$ "naturellement" si $x \geq 0$ et $1 - \frac{1}{1+e^{x}}$ si $x < 0$; ou encore en codant $\log(1+e^{-x})$ "naturellement" si $x \geq 0$ et $\log(1+e^{x}) - x$ si $x < 0$.

7. Simulations

L'idée des simulations est de tester les modèles utilisés ensuite sur les données réelles, mais en pouvant évaluer rigoureusement les performances puisque les paramètres sous-jacents sont connus car choisis pour générer les données. On cherchera alors à tester le schéma de prédiction suivant.

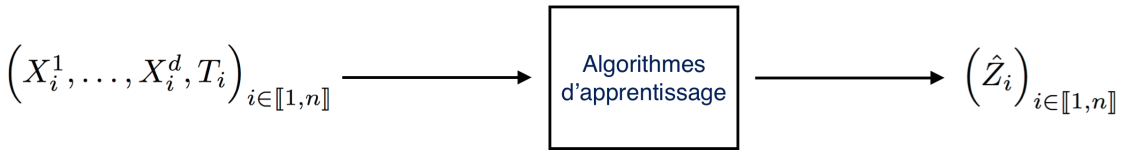

FIGURE 22: Schéma de prédiction

7.1. Génération des données

Nous allons alors tenter de comparer le cadre d'apprentissage supervisé explicité dans les sections précédentes, où on utilise une régression logistique pour prédire un label binaire, et le cadre du mélange de lois géométriques où on utilise l'algorithme EM. Il est clair que dans le cadre de l'EM, les $\hat{\mathbb{P}}(Z_i = 1)$ (qu'on notera simplement $\hat{Z}_i$) pourront être estimées naturellement par $1 - \pi_0(X_i)$ avec le β appris par l'algorithme. Pour ce qui est de la régression logistique, l'idée va alors être de suivre l'hypothèse initiale des cliniciens et de se rapprocher un maximum du procédé appliqué sur les données réelles, à savoir prendre un seuil s à 15 jours, créer des labels binaires $Y_i = \mathbb{1}_{\{T_i < s\}}$ et les assimiler aux Z_i.

Remarque 14 : Il serait intéressant à ce niveau de faire varier le seuil s et ceci sera à explorer davantage. Une autre façon de comparer les deux méthodes serait d'estimer la probabilité de retour avant 15 jours dans le cadre de l'EM, que l'on peut déduire de la distribution géométrique suivant la classe $\hat{Z}_i$ prédite. Le problème de cette façon de faire est qu'on aura alors en sortie l'une ou l'autre des deux probabilités de retour des deux lois géométriques avec les deux paramètres $\hat{p_0}$ et $\hat{p_1}$ appris, donc des prédictions binaires et non plus éclatées sur le segment $[0, 1]$. Utiliser l'AUC ne serait alors plus tout à fait pertinent pour comparer les deux méthodes.

Différentes manières de générer les données ont été essayées au cours du stage. Nous en présenterons ici une qui paraît se rapprocher le plus de nos données réelles.

On commence par choisir arbitrairement un vecteur des classes cachées suivant une loi de Bernoulli $\mathcal{B}(1-\pi_0)$ avec $\pi_0 = 0,872$ et pour $n = 1000$ séjours. Nous prendrons en fait pour les paramètres π_0, p_0 et p_1 ceux estimés sur les données réelles et résumés dans le tableau 11 de la section 8.2., on prendra donc $p_0 = 0,005$ et $p_1 = 0,511$.

Un fois les Z_i générés, on génère les $T_i \sim \mathcal{G}(p_{Z_i})$. Il ne reste alors plus qu'à générer les données d'apprentissage. On procédera de la façon suivante. On va travailler dans un espace de dimension $d = 100$ et on considèrera que seules les $a = 10$ premières features (choix arbitraire) sont informatives pour la tâche de prédiction pour se rapprocher des caractéristiques des données réelles. Le β que l'on va chercher à apprendre dans l'une ou l'autre des méthodes devra alors dans l'idéal avoir les a premiers coefficients non nuls et les autres nuls. Pour s'assurer qu'il y aura bien du signal à exploiter dans les données d'apprentissage, on les simulera de la façon suivante.

$$\forall i \in [\![1,n]\!], \begin{cases} \forall j \in [\![1,a]\!], \begin{cases} X_i^j | Z_i = 0 \sim \mathcal{N}(0,1) \\ X_i^j | Z_i = 1 \sim \mathcal{N}(0.5,1) \end{cases} \\ \forall j \in [\![a+1,d]\!], X_i^j \sim \mathcal{N}(0,1) \end{cases}$$

La constante 0,5 a été choisie volontairement petite pour que le problème ne soit pas "trop simple" et de nouveau pour mimer les caractéristiques des données réelles. En effet, on a pu constater que les différences dans les distributions des features conditionnellement à la classe dans le cadre de l'apprentissage supervisé étaient minimes. La différence entre les deux distributions $\mathcal{N}(0,1)$ et $\mathcal{N}(0.5,1)$ est légére, comme on peut le voir sur la figure suivante, mais suffisante pour permettre un apprentissage correct, comme nous allons le voir dans la suite.

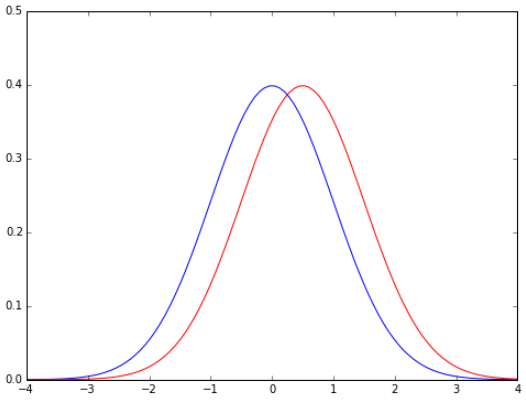

FIGURE 23: Densité de $\mathcal{N}(0,1)$ en bleu et $\mathcal{N}(0.5,1)$ en rouge

7.2. Comportement de l'approche régression logistique

Lorsqu'on lance alors une régression logistique avec une régularisation de type Elastic-Net, et après avoir séparé le jeu de données en un ensemble d'apprentissage et un ensemble de test, on obtient les résultats de prédictions sur le jeu de test résumés par la courbe ROC suivante dont l'AUC est de 0,69.

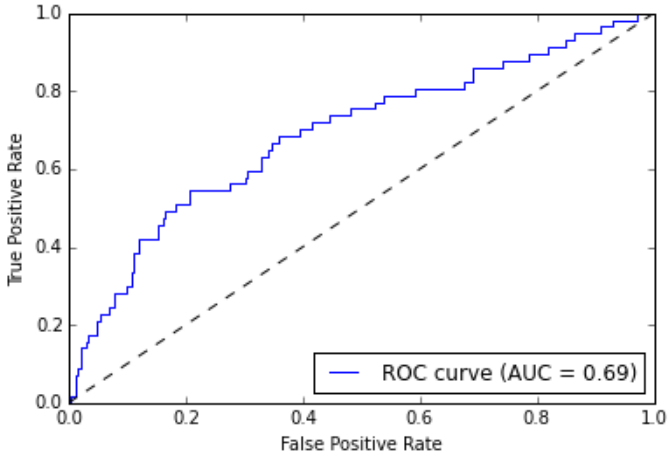

FIGURE 24: Courbe ROC pour la prédiction de la régression logistique

Si on regarde maintenant le $\hat{\beta}$ estimé par l'algorithme, on constate bien l'apprentissage marqué par des coefficients plus grands pour les 10 premiers, mais on constate qu'il reste du bruit avec de nombreux autres coefficients non nuls.

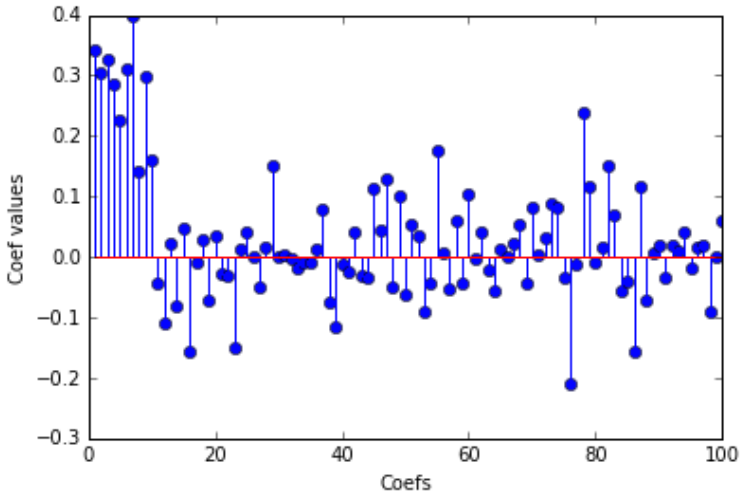

FIGURE 25: $\hat{\beta}$ appris par la régression logistique

7.3. Comportement de l'approche EM

En reprenant les mêmes ensembles d'apprentissage et de test, on lance alors l'algorithme EM avec une pénalisation Elastic-Net, qui a été codé de façon à pouvoir utiliser la cross-validation de scikit learn (avec donc une méthode *.fit* et une méthode *.predict*). On peut alors visualiser l'évolution de la log-vraissemblance calculée sur le jeu d'entraînement en bleu et sur le jeu de test en rouge au cours de l'apprentissage.

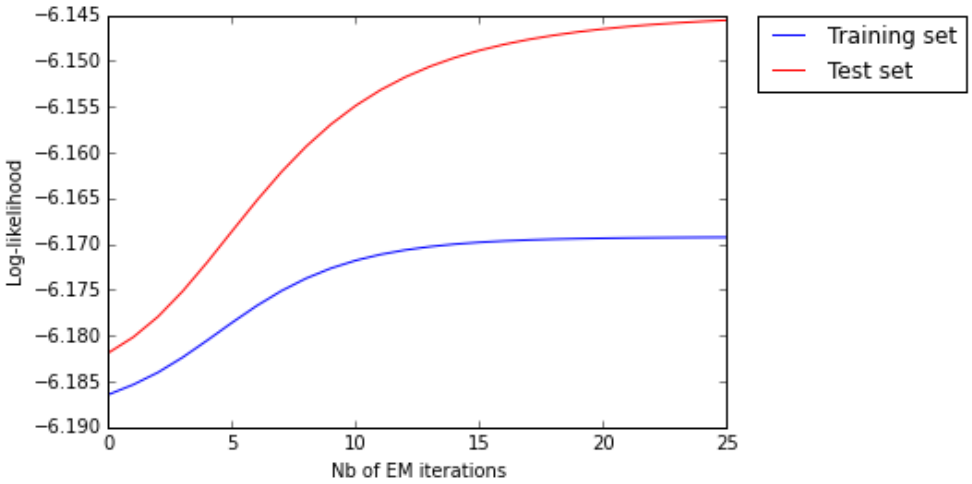

FIGURE 26: Evolution de la log-vraissemblance au cours des itérations

On obtient finalement la courbe ROC suivante pour les prédictions sur le jeu de test avec une AUC de 0,86 donc bien meilleure que pour la régression logistique.

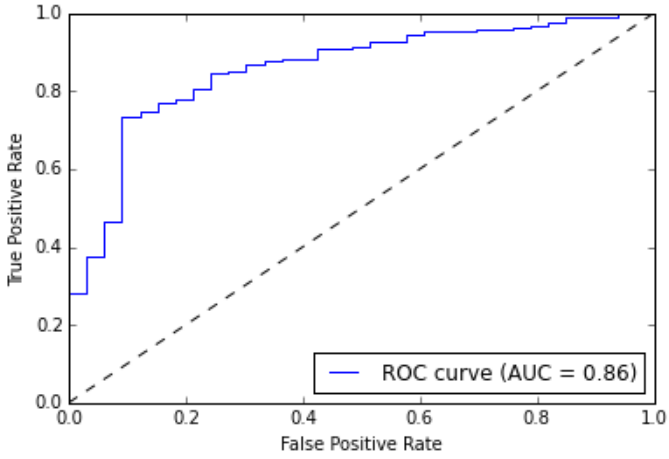

FIGURE 27: Courbe ROC pour la prédiction de l'EM

Si on observe mainenant le $\hat{\beta}$ estimé par l'algorithme, on constate également l'apprentissage avec des coefficients plus lourds pour les 10 premiers, et on remarque qu'il y a moins de bruits pour les autres coefficients qui sont en moyenne assez faibles.

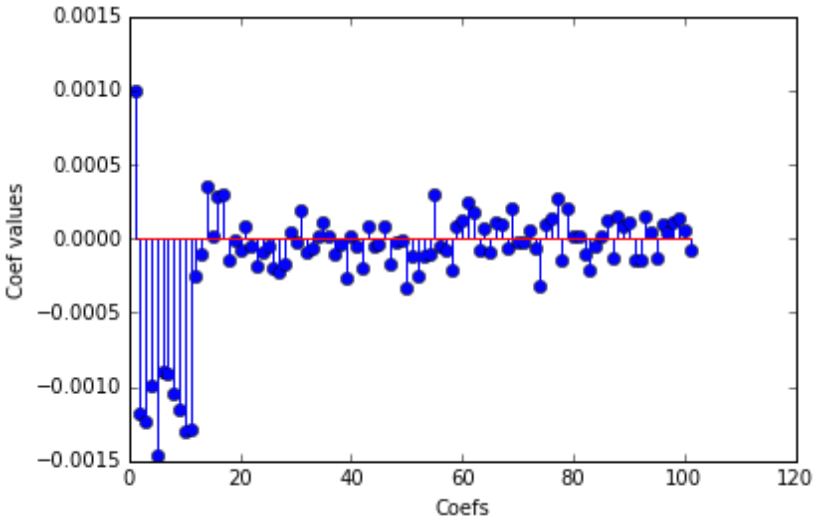

FIGURE 28: $\hat{\beta}$ appris par l'EM

Précisons que $\hat{\beta}$ est appris au signe près par l'algorithme EM, il ne faut alors pas interpréter le signe des coefficients. Notons aussi qu'une initialisation du $\hat{\beta}$ de l'EM avec les coefficients appris par la régression logistique a été essayée et que les résultats sont les mêmes que pour l'initialisation par le vecteur nul.

Lorsqu'on répète maintenant le procédé que l'on vient de décrire 100 fois de suite avec de nouvelles données générées à chaque itération, on obtient les résultats suivants avec une AUC moyenne de 0,75 pour la régression logistique et de 0,81 pour l'EM, ce qui confirme les meilleures performances de la seconde méthode.

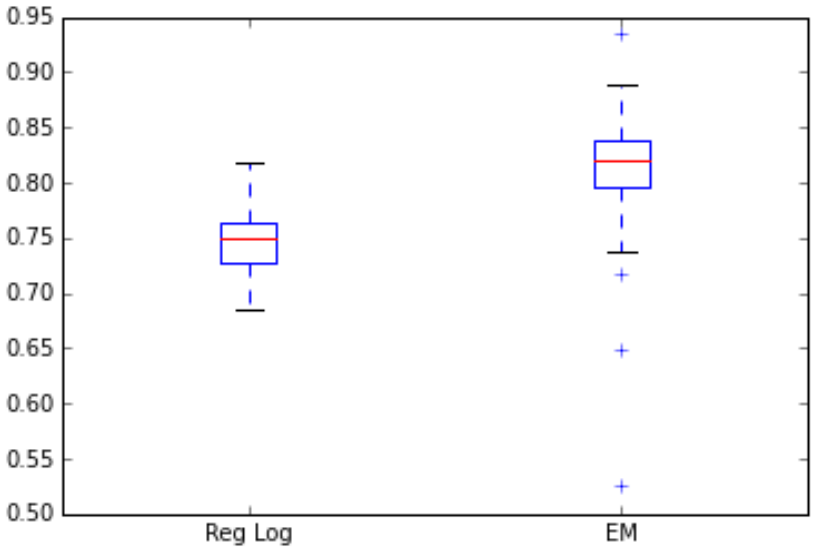

FIGURE 29: Comparaison des AUC des 2 méthodes pour 100 tests

8. Résultats sur les données réelles

Maintenant que les algorithmes ont été validés et ont fait leurs preuves sur les données simulées, on peut les tester sur les données réelles.

8.1. L'approche régression logistique

Commençons par faire une régression ridge dans l'espace de travail obtenu en sortie du préprocessing, en dimension 10546.
On divise alors l'ensemble du jeu de donnée en faisant un échantillonage stratifié afin de conserver les proportions des classes : on crée ainsi un ensemble d'apprentissage avec 70% des données, et un ensemble de validation inaccessible pendant la phase d'apprentissage avec les 30% restant. On fera sur le jeu d'apprentissage une 2-fold cross-validation pour choisir l'hyper-paramètre C_2, en procédant par grid-search en balayant un ensemble composé de 100 valeurs s'étalant de 10^{-5} à 10^3 selon une échelle logarithmique, avec $C_2 = \frac{1}{\lambda_2}$.
On peut visualiser l'apprentissage grâce à la heatmap qui suit représentant l'évolution de l'AUC (la moyenne sur les deux sous-ensembles de test créés à partir de l'ensemble d'apprentissage lors de la cross-validation) suivant la grille des hyper-paramètres, avec un dégradé de couleur représentant les plus faibles valeurs en bleu et les plus fortes en rouge.

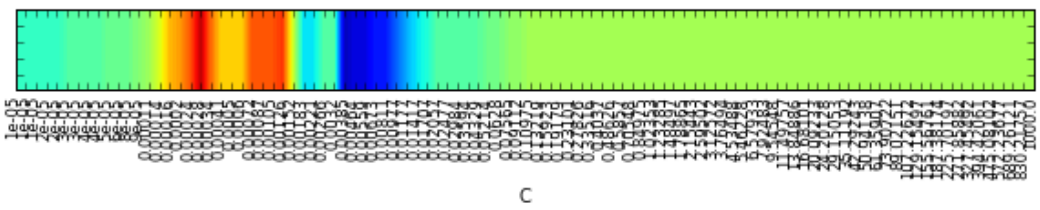

FIGURE 30: Fine-tunning du paramètre C_2

Remarque 15 : On vérifie visuellement ici la non convexité du problème d'optimisation en λ_2, en effet la

fonction $\lambda \mapsto \min_{\beta \in \mathbb{R}^d} \Big\{ f(\beta) + \lambda g(\beta) \Big\}$ n'est pas convexe en λ à cause du min.

On teste ensuite le prédicteur obtenu avec le meilleur paramètre qui est de $C_2 = 2,8 \times 10^{-4}$ sur le jeu de test et on obtient la courbe ROC suivante, avec un AUC de 0,81, ce qui est déjà un bon score.

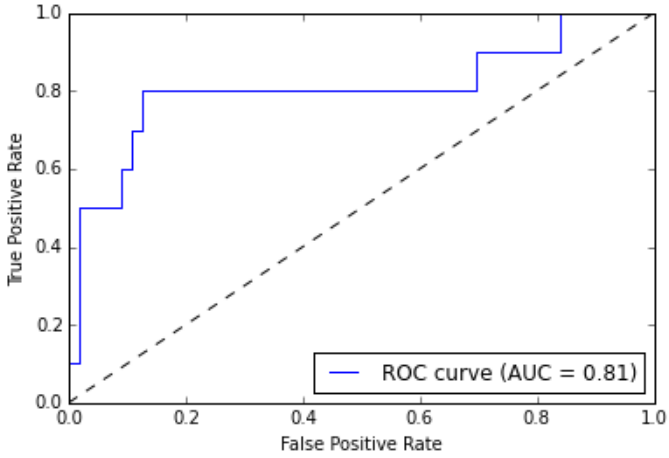

FIGURE 31: Courbe ROC pour la régression ℓ_2 dans l'espace de dimension 10546

Puis les résultats de prédiction en terme de précision et de recall sont résumés dans le tableau suivant.

Classe	Précision	Recall	Support
0	0,96	0,77	56
1	0,38	0,80	10
Moyenne/Total	0,87	0,77	66

TABLEAU 7 : Résultats de prédiction

Ainsi les résultats sont relativement bons et peuvent être améliorés en étant plus précis sur la prédiction de la classe positive, où on fait 60% de prédictions de Faux Positifs.
On peut alors visualiser les coefficients du vecteur de poids appris qui ont été triés, avec à gauche les poids réels et à droite leurs valeurs absolues.

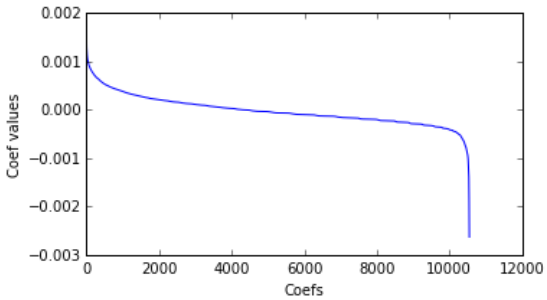

FIGURE 32: Coefficients de $\hat{\beta}_{ridge}$ triés

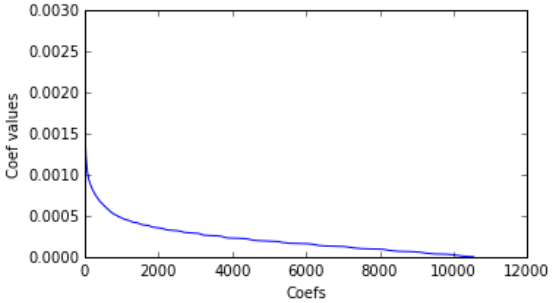

FIGURE 33: Coefficients de $|\hat{\beta}_{ridge}|$ triés

On peut interpréter la figure de droite comme la courbe d'importance décroissante des features de l'espace. Sa forme nous incite à tenter de réduire l'espace de très grande dimension en un espace de bien plus faible dimension, car dans un tel espace, la fonction optimale à apprendre devrait être plus régulière et gagner en pouvoir de généralisation dans un espace de complexité plus faible.

Mais avant de réduire encore la dimension de l'espace de travail, observons sur la figure suivante les probabilités de prédiction sur le jeu de test (composé de 66 séjours). On a en vert les probabilités donnant lieu à une prédiction binaire correcte et en rouge celles qui engendrent des erreurs. On constate que globalement, l'ensemble des probabilités est concentré autour du seuil de décision de 0,5 et l'algorithme ne semble pas très fiable, même si les performances restents correctes.

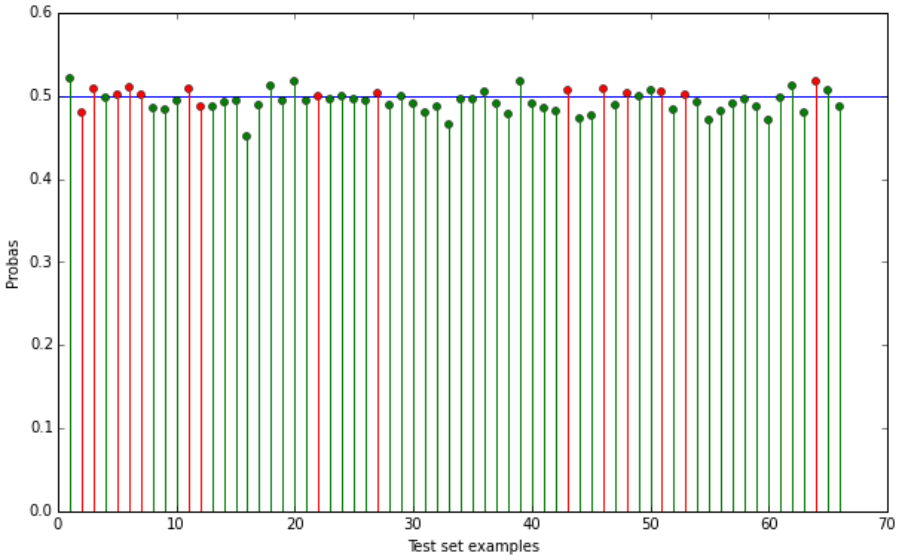

FIGURE 34: Probabilités prédites sur le jeu de test

Nous allons maintenant réaliser la même procédure, au détail près que nous changerons de méthode de régularisation avec une régression lasso de façon à profiter de ses propriétés de sélection de variables. Le meilleur hyper-paramètre $C_1 = \frac{1}{\lambda_1}$ obtenu est de 0,49 et on obtient les précisions et recalls suivants suivis par la courbe ROC correspondante évaluée sur le même jeu de test que précédemment, avec une AUC de $0,88$ qui améliore les performances précédentes.

Classe	Précision	Recall	Support
0	0,98	0,82	56
1	0,47	0,90	10
Moyenne/Total	0,90	0,83	66

TABLEAU 8 : Résultats de prédiction

On constate en effet que globalement, les scores ont été améliorés.

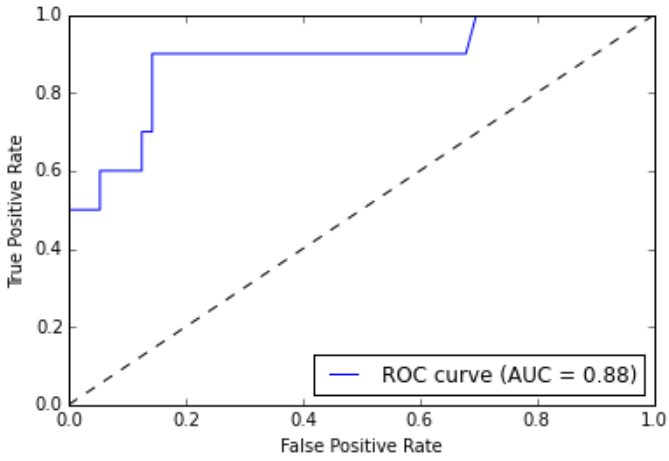

FIGURE 35: Courbe ROC pour la régression ℓ_1 dans l'espace de dimension 10546

Quand on regarde maintenant les coefficients de $|\hat{\beta}_{lasso}|$ non nuls et triés, on obtient le graphe suivant, où on a renseigné le nom des features (avec les valeurs des quartiles concernés) pour les interprétations cliniques. On a alors sélectionné de cette façon 30 features, avec un mélange de variables biologiques, de paramètres vitaux, de variables seules ou en association. Les features sélectionnées sont listées ci-dessous.

previous_visit (680, 2412] ; (PA max [mmHg] gradient 15 (0.0468, 0.979],PA min [mmHg] gradient 4 [-5.0864, 0.142]) ; (PA max [mmHg] gradient 15 (0.0468, 0.979],PA min [mmHg] point 2 [40.41, 64.562]) ; (Frquence cardiaque [bpm] gradient 10 (0.773, 7.252],Frquence respiratoire [mvt/min] gradient 11 (0.145, 1.905]) ; (Frquence cardiaque [bpm] gradient 10 (0.773, 7.252],PA max [mmHg] gradient 24 (0.713, 7.0087]) ; (Temprature [C] point 26 (37.0982, 39.0774],Frquence respiratoire [mvt/min] gradient 11 (0.145, 1.905]) ; (43. (01ALAT) ALAT [6, 17.5],PA max [mmHg] point 24 (107.362, 115.974]) ; (Temprature [C] point 18 (37.0659, 38.892],Temprature [C] gradient 10 (0.0246, 0.212]) ; (PA max [mmHg] gradient 24 (0.713, 7.0087],Douleur EVA gradient 11 (0.00876, 1.108]) ; (PA min [mmHg] gradient 3 [-4.36, 0.201],previous_visit (680, 2412]) ; (Temprature [C] gradient 2 (-0.0293, 0.000523],Douleur EVA point 19 [0, 1.879]) ; (duration [0, 3.25],Frquence respiratoire [mvt/min] point 1 (17.751, 19.217]) ; (Douleur EVA point 19 [0, 1.879],previous_visit (680, 2412]) ; (Douleur EVA point 13 (4.0109, 7.676],PA min [mmHg] gradient 3 [-4.36, 0.201]) ; (Frquence cardiaque [bpm] gradient 10 (0.773, 7.252],PA max [mmHg] gradient 25 (1.162, 18.0914]) ; (Frquence respiratoire [mvt/min] gradient 11 (-0.00955, 0.0383],previous_visit (680, 2412]) ; Temprature [C] gradient 0 [-0.85, -0.0846] ; (PA max [mmHg] gradient 24 (0.713, 7.0087],PA min [mmHg] point 2 [40.41, 64.562]) ; (43. (01ALAT) ALAT [6, 17.5],Temprature [C] gradient 2 (-0.0293, 0.000523]) ; (Temprature [C] gradient 10 (0.0246, 0.212],Dbit O2 [L/min] gradient 22 [-0.478, 0]) ; (PA max [mmHg] gradient 24 (0.713, 7.0087],Temprature [C] point 20 (37.076, 38.935]) ; PA max [mmHg] gradient 8 [-8.663, -1.0558] ; (PA max [mmHg] gradient 25 [-16.825, -1.743],PA min [mmHg] gradient 4 [-5.0864, 0.142]) ; (PA min [mmHg] gradient 1 [-12.394, -0.479],PA min [mmHg] gradient (0.201, 4.694]) ; (43. (01ALAT) ALAT [6, 17.5],Frquence respiratoire [mvt/min] point 1 [10.477, 17.751]) ; (PA max [mmHg] gradient 13 (0.948, 8.292],PA min [mmHg] gradient 3 [-4.36, 0.201]) ; (duration [0, 3.25],Temprature [C] gradient 0 (0.0474, 0.134]) ; (duration [0, 3.25],Douleur EVA gradient 11 (-0.119, -0.0457]) ; (18. (01PT) Protines (76, 85],PA max [mmHg] point 24 (107.362, 115.974]) ; (Frquence cardiaque [bpm] gradient 10 (0.773, 7.252],PA max [mmHg] point 25 [85.973, 109.589])

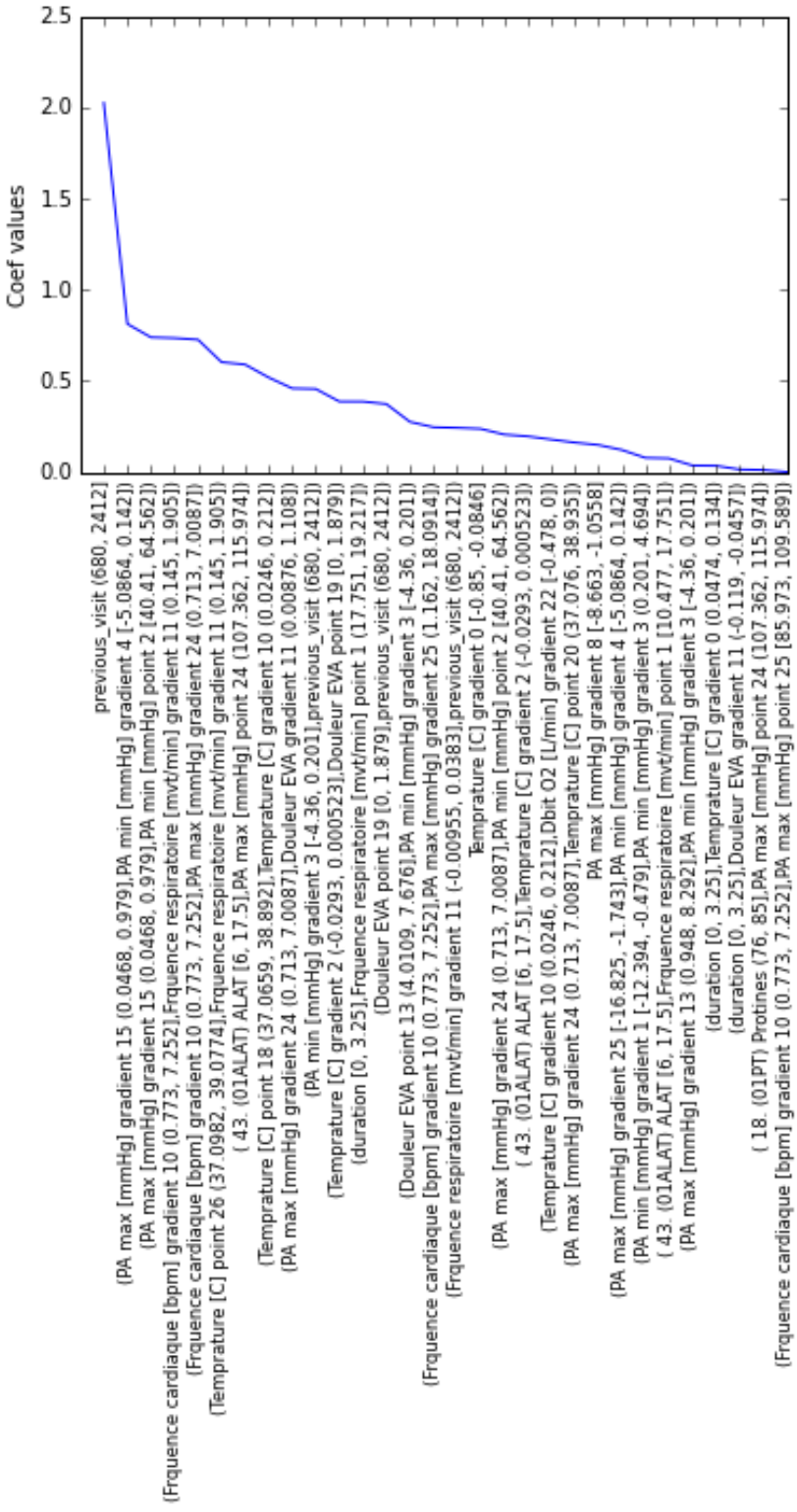

FIGURE 36: Coefficients de $|\hat{\beta}_{lasso}|$ non nuls triés

L'idée est alors de relancer la même procédure de régression ℓ_2 que celle essayée précédement, mais dans l'espace engendré par les features sélectionnées par le support du précédent lasso, ce qui constituera notre meilleur résultat parmi tous les essais réalisés. En utilisant la même grille pour l'hyper-paramètre C_2, on obtient les courbes d'apprentissage suivantes où on observe en bleu l'évolution de l'AUC moyen calculé sur les fold de test (on fait une cross-validation avec 2 fold uniquement car nous disposons de peu de données) et en rouge sur le jeu de validation extérieur (celui des 66 séjours).

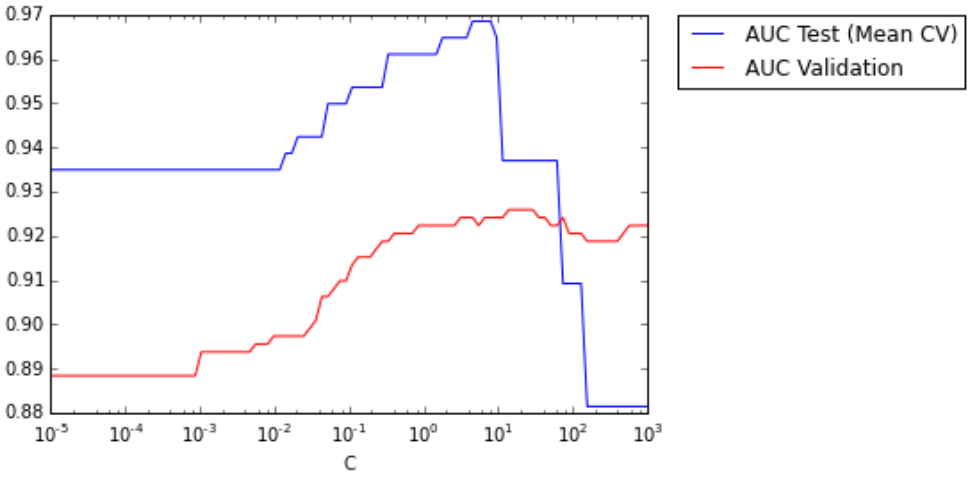

FIGURE 37: Courbe d'apprentissage pendant la grid-search de la régression ridge en dimension 30

Remarque 16 : Les deux courbes devraient en théorie avoir une asymptote horizontale d'équation $y = 0,5$ en $\pm\infty$. Le fait que les valeurs atteintes à droite et à gauche soient plus élevées que 0,5 est dû exclusivement à la taille des échantillons particulièrement petite, ce qui donne lieu à ces artéfacts. C'est donc plus ici l'allure des courbes qu'on interprétera, et on constate bien un apprentissage correct avec des maxima très proches pour les deux courbes.

On obtient alors les coefficients triés suivants pour $\hat{\beta}_{ridge}$ avec le meilleur paramètre C_2 sélectionné à $4,5$, donc peu de régularisation.

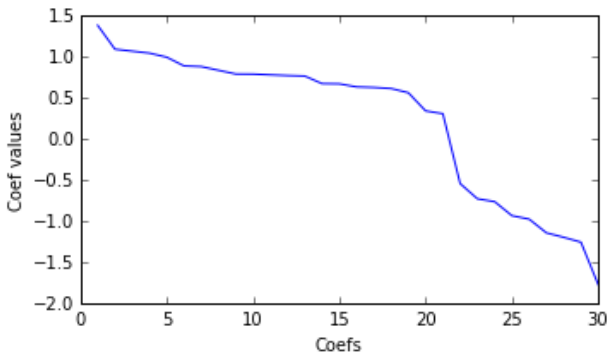

FIGURE 38: Coefficients de $\hat{\beta}_{ridge}$ triés

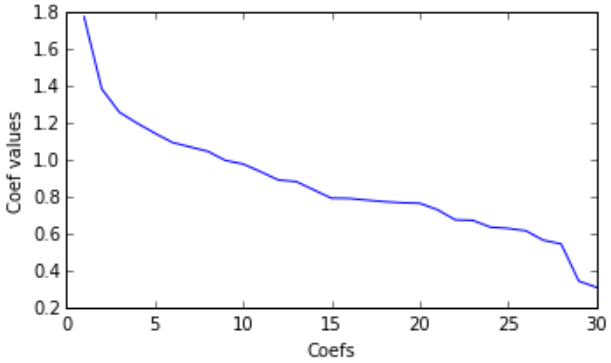

FIGURE 39: Coefficients de $|\hat{\beta}_{ridge}|$ triés

Puis les résultats de prédiction en terme de précision et de recall sont résumés dans le tableau suivant, suivis par la courbe ROC correspondante avec un AUC cette fois de $0,92$, ce qui constitue un très bon résultat.

Classe	Précision	Recall	Support
0	0,96	0,89	56
1	0,57	0,80	10
Moyenne/Total	0,90	0,88	66

TABLEAU 9 : Résultats de prédiction

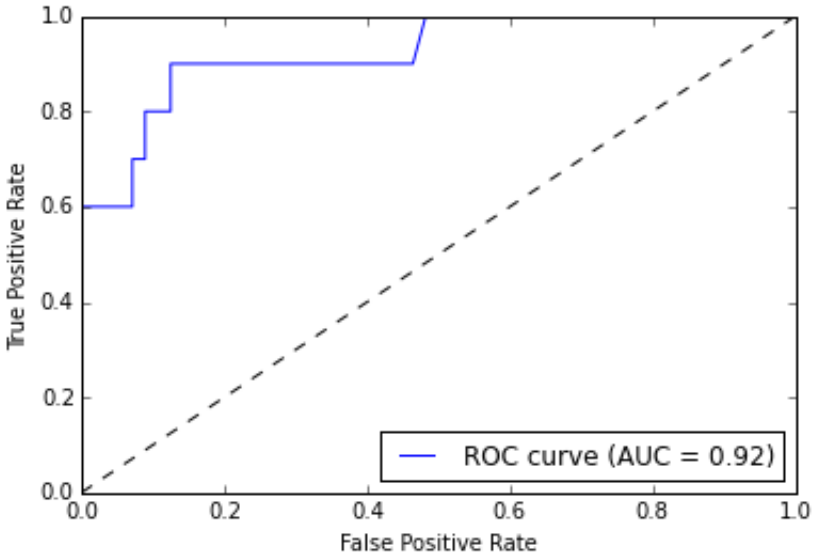

FIGURE 40: Courbe ROC pour la régression ℓ_2 dans l'espace de dimension 30

On a donc encore amélioré les résultats en terme de prédiction binaire, notamment avec une meilleure AUC, et on a surtout gagné en confiance avec une fiabilité du prédicteur beaucoup plus puissance, ce qu'on peut observer avec la représentation des probabilités prédites sur le jeu de test suivantes.

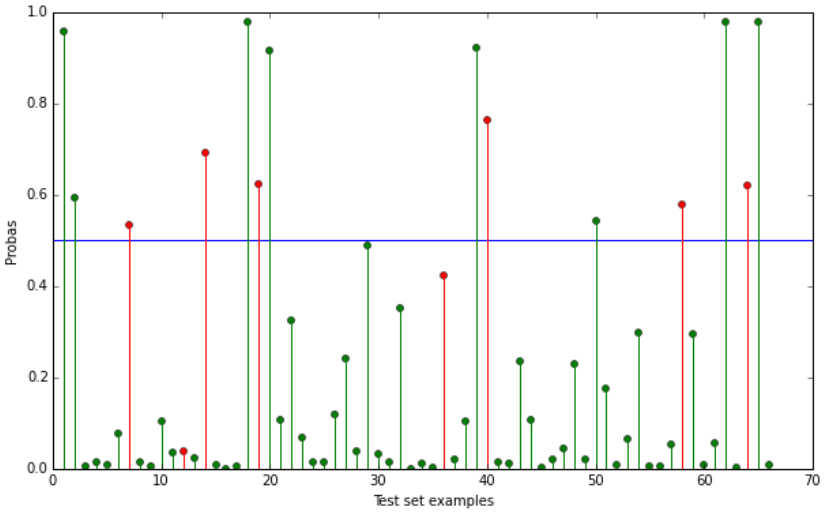

FIGURE 41: Probabilités prédites sur le jeu de test

Remarque 17 : Une interprétation détaillée avec les cliniciens et une visualisation plus claire des résultats est prévue début octobre. En pratique, il restera pour cette partie à essayer la régularisation Elastic-Net, puis une sélection de feature par l'adaptive lasso.

8.2. L'approche EM

Avant toute chose, une première question à se poser va être celle de l'initialisation des paramètres du modèle, et le problème d'optimisation étant non convexe, une bonne initialisation est importante. On pourra prendre par exemple prendre le vecteur nul pour β. Mais pour p_0 et p_1, nous utiliserons les estimateurs du maximum de vraissemblance sur le modèle de mélange en ne prenant que les temps de retours.

On a donc affaire au modèle simplifié suivant

$$\forall t \in \mathbb{N}^*, \forall z \in \{0,1\}, \mathbb{P}(T_i = t | Z_i = z) = \mathcal{G}(p_z) \text{ et } \mathbb{P}(Z_i = 1) = 1 - \pi_0$$

tel que

$$\forall t \in \mathbb{N}^*, \mathbb{P}(T_i = t) = (1-p_0)^{t-1} p_0 \pi_0 + (1-p_1)^{t-1} p_1 (1-\pi_0).$$

Les paramètres du modèle sont résumés dans le vecteur $\theta = (p_0, p_1, \pi_0)$ et la log-vraissemblance du modèle s'écrit

$$\ell(\theta) = \tfrac{1}{n} \sum_{i=1}^n \log \Big((1-p_0)^{T_i - 1} p_0 \pi_0 + (1-p_1)^{T_i - 1} p_1 (1-\pi_0) \Big).$$

La log-vraissemblance complétée par les variables latentes Z_i s'écrit quant à elle

$$\ell_c(\theta) = \tfrac{1}{n} \sum_{i=1}^n \Big((1-Z_i) \Big((T_i - 1) \log(1-p_0) + \log p_0 + \log \pi_0 \Big) + Z_i \Big((T_i - 1) \log(1-p_1) + \log p_1 + \log(1-\pi_0) \Big) \Big).$$

Posons alors $\mathcal{Q}_n(\theta, \theta^{(l)}) = \mathbb{E}_{\theta^{(l)}}[\ell_c(\theta) \mid \mathbf{T}]$.

Or

$$\mathbb{E}_{\theta^{(l)}}[Z_i \mid T_i] = \mathbb{P}_{\theta^{(l)}}(Z_i = 1 \mid T_i) = \frac{(1-p_1^{(l)})^{T_i - 1} p_1^{(l)} (1-\pi_0^{(l)})}{(1-p_1^{(l)})^{T_i - 1} p_1^{(l)} (1-\pi_0^{(l)}) + (1-p_0^{(l)})^{T_i - 1} p_0^{(l)} \pi_0^{(l)}} = q_i^{(l)}.$$

Ce qui donne

$$\mathcal{Q}_n(\theta, \theta^{(l)}) = \frac{1}{n} \sum_{i=1}^n \Big((1 - q_i^{(l)}) \Big((T_i - 1) \log(1-p_0) + \log p_0 + \log \pi_0 \Big) + q_i^{(l)} \Big((T_i - 1) \log(1-p_1) + \log p_1 + \log(1-\pi_0) \Big) \Big)$$

Dans l'étape M de l'algorithme, on doit maximiser $\theta \mapsto \mathcal{Q}_n(\theta, \theta^{(l)})$. La mise à jour de $\theta = (p_0, p_1, \pi_0)$ consiste donc simplement à annuler les dérivées partielles de $\mathcal{Q}_n(\cdot, \theta^{(l)})$, ce qui amène aux mises à jour suivantes pour θ :

$$p_0^{(l+1)} = \frac{n - \sum_{i=1}^n q_i^{(l)}}{\sum_{i=1}^n (1 - q_i^{(l)}) T_i}, \quad p_1^{(l+1)} = \frac{\sum_{i=1}^n q_i^{(l)}}{\sum_{i=1}^n q_i^{(l)} T_i} \text{ et } \pi_0^{(l+1)} = 1 - \tfrac{1}{n} \sum_{i=1}^n q_i^{(l)}$$

Commençons par tester l'implémentation réalisée pour s'assurer que tout fonctionne correctement. On va alors simuler $n = 1000$ temps de retour selon un mélande de loi géométrique, en prenant $\pi_0 = 0,8$. On génére d'abord la classe cachée $\forall i \in [\![1, n]\!], Z_i \sim \mathcal{B}(\pi_0)$ loi de Bernoulli de paramètre π_0, puis les temps de retour $\forall i \in [\![1, n]\!], T_i \sim \begin{cases} \mathcal{G}(p_1) \text{ loi géométrique de paramètre } p_1 \text{ si } Z_i = 1 \\ \mathcal{G}(p_0) \text{ loi géométrique de paramètre } p_0 \text{ si } Z_i = 0 \end{cases}$, avec $p_0 = 0,1$ et $p_1 = 0,5$.

On peut observer la distribution des temps sur le graphe suivant.

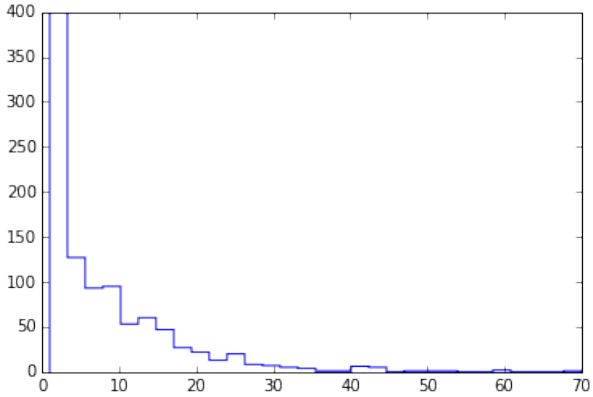

FIGURE 42: Distribution des temps du mélange simulé

On lance ensuite l'algorithme EM et on observe sur les courbes suivante à gauche l'évolution de la log-vraissemblance et à droite l'évolution de l'erreur quadratique $\|\theta - \hat{\theta}\|_2$ au cours des différentes itérations.

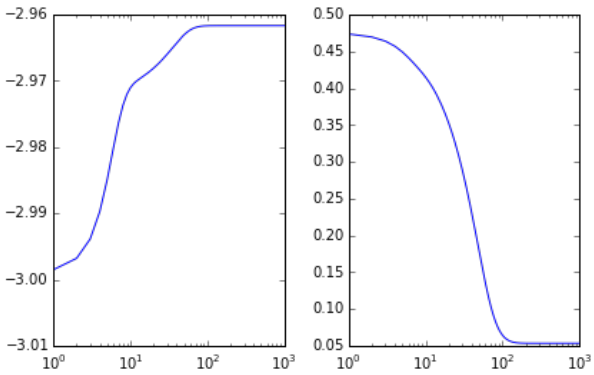

FIGURE 43: Evolution de la log-vraissemblance et de l'erreur quadratique en fonction des itérations en simumlation

On a initialisé ici π^{ini} à 0,5, puis on a tiré des Z_i selon une Bernoulli de paramètre 0,5, et pris les estimateurs du maximum de vraissemblance pour les p_k pour $k \in \{0, 1\}$ avec $p_k^{ini} = \frac{1}{\sum_{i=1}^n \mathbb{1}_{\{Z_i=k\}}} \sum_{i=1}^n T_i \mathbb{1}_{\{Z_i=k\}}$.
On peut alors constater l'efficacité de l'algorithme en comparant les vrais paramètres et les paramètres estimés dans le tableau suivant.

Paramètres	p_0	p_1	π
θ	0,1	0,5	0,8
$\hat{\theta}$	0,101	0,491	0,747

TABLEAU 10 : Résultats de prédiction sur les simulations

On lance alors le même algorithme mais cette fois sur les temps réels de notre jeu de données en prenant garde d'ajouter 1 à tous les temps pour respecter le support des lois géométriques. On a la courbe de la log-vraissemblance suivante lors de l'apprentissage au cours des itérations.

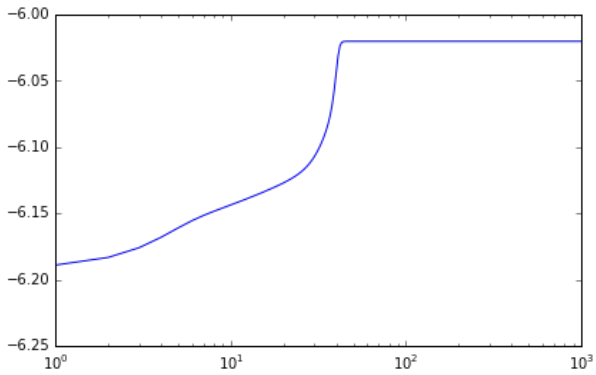

FIGURE 44: Evolution de la log-vraissemblance au cours de l'apprentissage sur les données réelles

On constate bien la stricte croissance et la convergence garanties par la théorie. On obtient les paramètres estimés suivants.

Paramètres	p_0	p_1	π
$\hat{\theta}$	0,005	0,511	0,872

TABLEAU 11 : Résultats de prédiction sur les vraies données

Et lorsqu'on simule un mélange de lois géométriques avec les paramètres estimés, on obtient la distribution de droite suivante, avec à sa gauche la distribution des temps réels de retour.

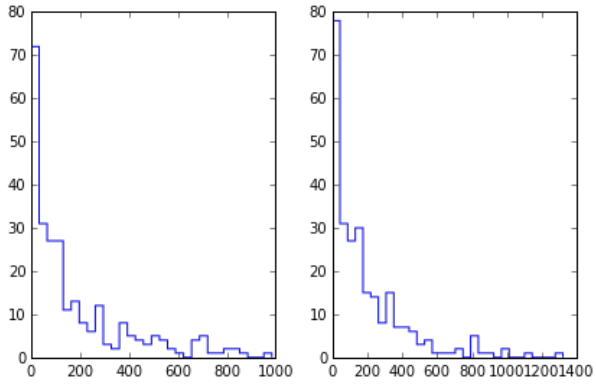

FIGURE 45: Comparaison des distributions de temps de retour réels et simulés après estimation

On constate donc qu'on retrouve bien l'allure de la distribution réelle, et on se servira de ces paramètres dans l'initialisation des paramètres pour l'EM à suivre.

Comme celà a déjà été évoqué, on est confronté en pratique au problème de la censure dès lors qu'on tente d'utiliser les données réelles avec le modèle de mélange. En effet, comment renseigner le label T_i pour l'ensemble des derniers séjours des patients, ou tout simplement pour l'ensemble des patients n'ayant qu'un seul séjour dans la base de donnée ?

L'idée est alors de mettre en jeu trois nouvelles variables aléatoires. Pour tout $i \in [\![1, n]\!]$,
$\begin{cases} R_i \in \mathbb{N}^* \ v.a. \text{ modélisant le nombre de jours réel entre le séjour } i \text{ et le prochain séjour} \\ C_i \in \mathbb{N}^* \ v.a. \text{ modélisant le nombre de jours entre le séjour } i \text{ et la limite supérieure d'observation} \\ \delta_i = \mathbb{1}_{\{R_i \leq C_i\}} \ v.a. \text{ modélisant la présence d'une censure pour le temps } T_i \end{cases}$

On appellera la $v.a.$ C_i variable de censure, et on re-définit la variable T_i des sections précédentes comme $T_i = R_i \wedge C_i$. Ainsi, T_i prend comme valeur soit la vraie valeur de retour du patient i si R_i est dans la fenêtre d'observation du jeu de donnée (donc avant 2015), sinon la valeur de C_i étant le nombre de jours écoulés entre le séjour i et le dernier jour observable de la base de donnée.
Et les labels ne sont désormais plus simplement les temps T_i mais les couples $(T_i, \delta_i)_{i \in [\![1,n]\!]}$ de $\mathbb{N}^* \times \{0, 1\}$, avec $\delta_i = 1$ si le temps observé T_i n'est pas censuré et 0 sinon.

Afin d'adapter l'algorithme EM décrit précédemment, nous aurons besoin de deux hypothèses d'indépendance qui simplifient beaucoup les choses et permettent les calculs à venir.

Hypothèse 4 : On suppose que $\forall i \in [\![1, n]\!], R_i \perp\!\!\!\perp C_i$ conditionnellement à Z_i et X_i.

Hypothèse 5 : On suppose pour tout i dans $[\![1, n]\!]$, la loi de C_i est indépendante de Z_i et X_i.

Il suffit maintenant d'écrire la log-vraissemblance complétée du nouveau modèle sous ces hypothèses (dans la suite, tout est conditionné par les X_i).
En notant G la fonction de répartition de C ; $\overline{G} = 1 - G$; $\forall k \in \{0, 1\}, \forall t \in \mathbb{N}^*, \mathbb{P}(R_i = t | Z_i = k) = p_k(t)$, F_k la fonction de répartition et $\overline{F_k} = 1 - F_k$, on a alors pour tout $t \in \mathbb{N}^*$,

$$\begin{cases} \mathbb{P}(T_i = t, \delta_i = 1 | Z_i = 1) = \mathbb{P}(R_i = t, C_i \geq t | Z_i = 1) = p_1(t)\overline{G}(t^-) \\ \mathbb{P}(T_i = t, \delta_i = 1 | Z_i = 0) = \mathbb{P}(R_i = t, C_i \geq t | Z_i = 0) = p_0(t)\overline{G}(t^-) \\ \mathbb{P}(T_i = t, \delta_i = 0 | Z_i = 1) = \mathbb{P}(C_i = t, R_i \geq t | Z_i = 1) = g(t)\overline{F_1}(t^-) \\ \mathbb{P}(T_i = t, \delta_i = 0 | Z_i = 0) = \mathbb{P}(R_i = t, C_i \geq t | Z_i = 0) = g(t)\overline{F_0}(t^-) \end{cases}$$

Puis pour tout $z \in \{0, 1\}$,

$$\begin{cases} \mathbb{P}(T_i = t, \delta_i = 1, Z_i = z) = p_1(t)^z \overline{G}(t^-)^z p_0(t)^{1-z} \overline{G}(t^-)^{1-z} \pi_0(X_i)^{1-z}(1 - \pi_0(X_i))^z \\ \mathbb{P}(T_i = t, \delta_i = 0, Z_i = z) = g(t)^z \overline{F_1}(t^-)^z g(t)^{1-z} \overline{F_0}(t^-)^{1-z} \pi_0(X_i)^{1-z}(1 - \pi_0(X_i))^z \end{cases}$$

D'où, en notant $\begin{cases} \mathcal{T} = (T_1, \dots, T_n) \\ \mathcal{Z} = (Z_1, \dots, Z_n) \\ \Delta = (\delta_1, \dots, \delta_n) \end{cases}$,

$$\begin{aligned} \ell_n^c(\theta, \mathcal{T}, \Delta, \mathcal{Z}) = \frac{1}{n} \sum_{i=1}^{n} \delta_i \Big\{ Z_i [\log(1 - \pi_0(X_i)) + \log(p_1(T_i))] + (1 - Z_i)[\log(\pi_0(X_i)) + \log(p_0(T_i))] + \log(\overline{G}(T_i^-)) \Big\} \\ + (1 - \delta_i) \Big\{ Z_i [\log(1 - \pi_0(X_i)) + \log(\overline{F_1}(T_i^-))] + (1 - Z_i)[\log(\pi_0(X_i)) + \log(\overline{F_0}(T_i^-))] + \log(g(T_i)) \Big\} \end{aligned}$$

Notons que mis à part les termes en G et g qui n'interviendront pas dans la mise à jour des paramètres lors de l'EM, le terme en facteur du δ_i est le même que celui qu'on avait dans la log-vraissemblance complétée sans prendre en compte la censure (ce qui se comprend bien car dans ce cas δ_i vaut toujours 1), et le terme en facteur de $(1 - \delta_i)$ est nouveau.

L'implémentation étant en cours pour intégrer la censure au code existant, nous aurons rapidement les résultats de l'EM sur les données réelles.

9. Conclusion

Ainsi, il reste encore de nombreuses pistes à explorer dont les idées ont été évoquées tout au long de ce rapport. Mais les premiers résultats obtenus, tant sur les données simulées que sur les données réelles, sont très encourageants.
La petite taille de la cohorte de patients dont nous disposons est un handicap certain lorsqu'il s'agit de valider de façon solide les modèles. Aussi pourrons nous essayer de valider les prédicteurs sur des bases de données extérieures.

J'ai été absolument ravi sur tous les plans au cours de ces 5 mois de stage qui sont passés très vite, et c'est avec plaisir et hâte que ma thèse débutera en octobre à l'école doctorale de sciences mathématiques de Paris Centre. Cette thèse aura pour but d'introduire de nouvelles méthodes pour la prise en compte de variables longitudinales et d'en étudier les propriétés théoriques. Nous travaillerons alors notamment sur une cohorte très importante : la cohorte Artemis composée de plus de 30000 patients hypertendus suivis pendant au moins 5 ans.
Les données de cette base auront des caractéristiques proches de celles de ce stage : un grand nombre de covariables longitudinales et une grande variabilité inter-individus dans les mesures (nombre et espacements des rélevés), mais avec cette fois un caractère fortement "multi-class" du problème supervisé et un grand nombre d'individus.
L'idée sera de se focaliser davantage sur le problème de non-alignement des temps de mesures (ou de censure) de ces variables et sur la variabilité des temps. Différents algorithmes d'apprentissage seront alors étudiés dans ce cadre de travail, comme les SVM, la régression logistique "multi-class" ou la régression de Cox "multi-task" avec pénalisation group-lasso pour les variables longitudinales, ou encore des méthodes de type "forêts aléatoires". Enfin, eu égard à la taille de la base de données, le problème de "passage à l'échelle" (parallélisation, algorithmes stochastiques) des algorithmes proposés sera étudié.

J'ai aussi été ravi d'enrichir ma culture en apprentissage statistique pendant ces quelques mois d'été, avec ma présence à tous les séminaires SMILE, ou encore ma participation à la conférence internationale de Machine Learning (ICML) en tant que volontaire où j'ai pu échanger avec des étudiants du monde entier, et interagir avec des chercheurs comme par exemple Lian Wenzhao qui a présenté un papier [13] où l'objet était de prédire des temps de retour à l'hôpital, en se basant uniquement sur les dates des séjours. Les performances étaient plutôt bonnes et cela fait écho à la feature "previous_visit" qui resort en premier de nos modèles.

Enfin, ce stage m'aura permi de faire des progrès techniques ; avec par exemple la maîtrise du langage de programmation python, du format JSON et des avantages qu'il apporte, ou encore en programmation orientée objet.

10. Annexes

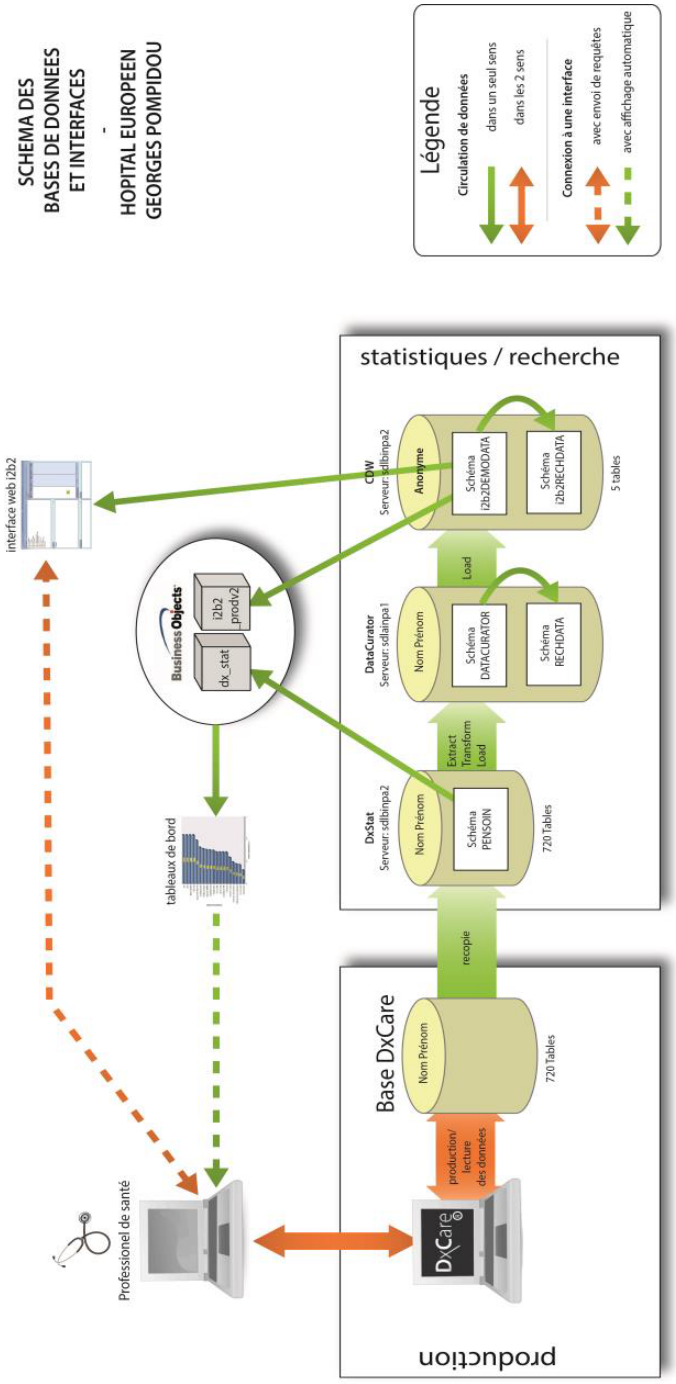

FIGURE 46: Organisation des données à l'HEGP

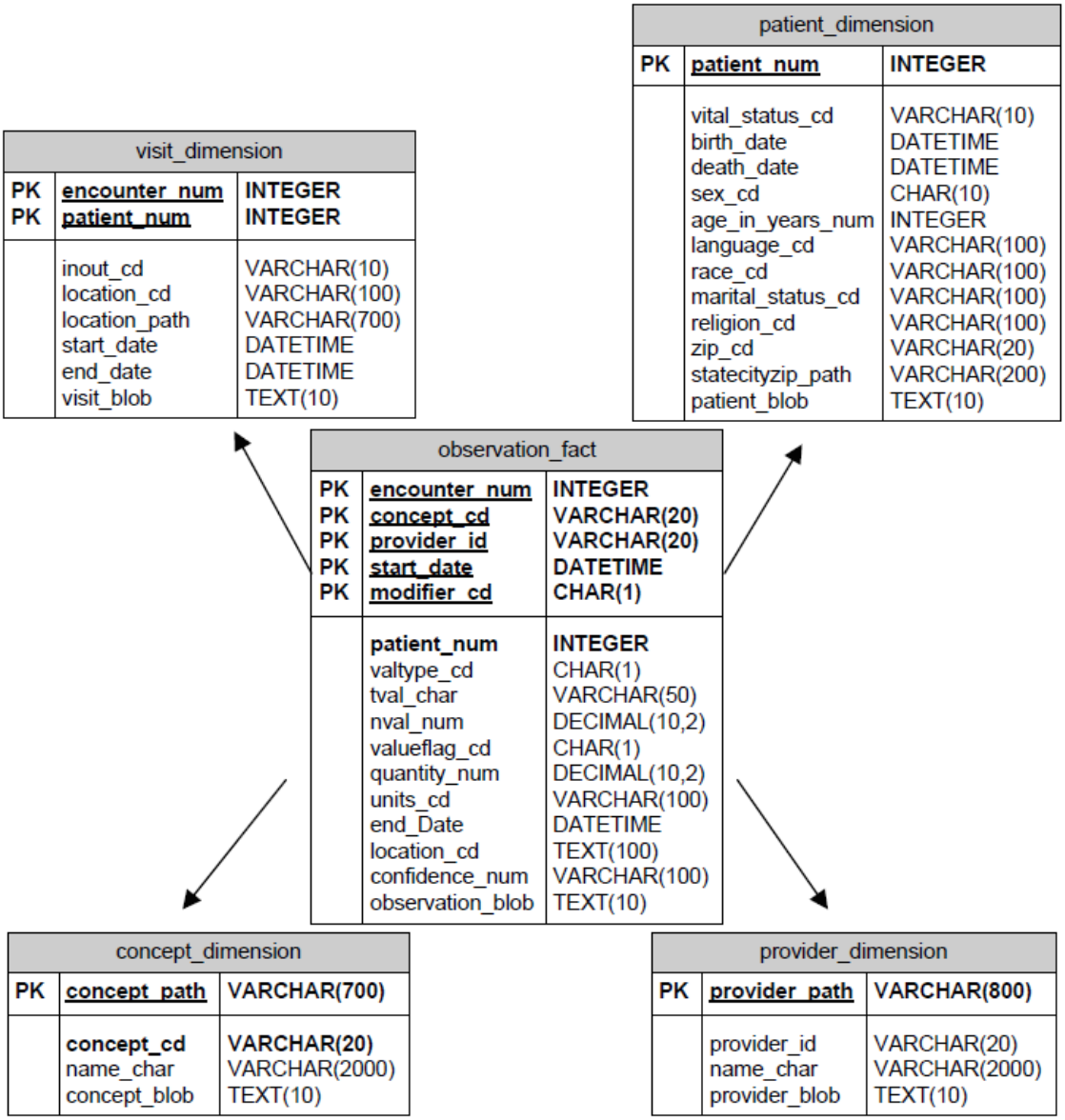

FIGURE 47: Diagramme de classes de l'entrepôt I2B2

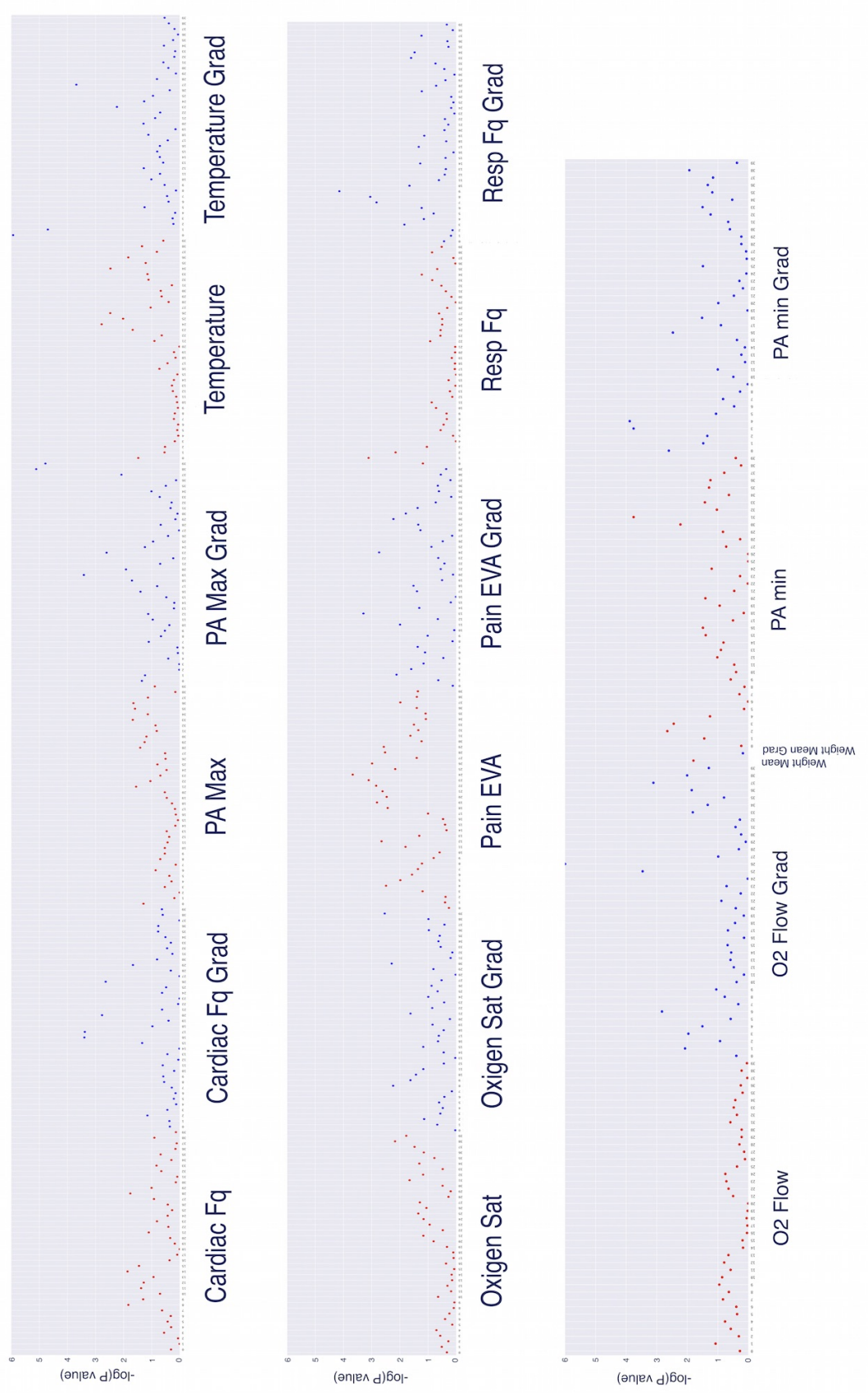

FIGURE 48: Tests de Mann-Whitney-Wilcoxon conditionnellement à la classe pour les paramètres vitaux

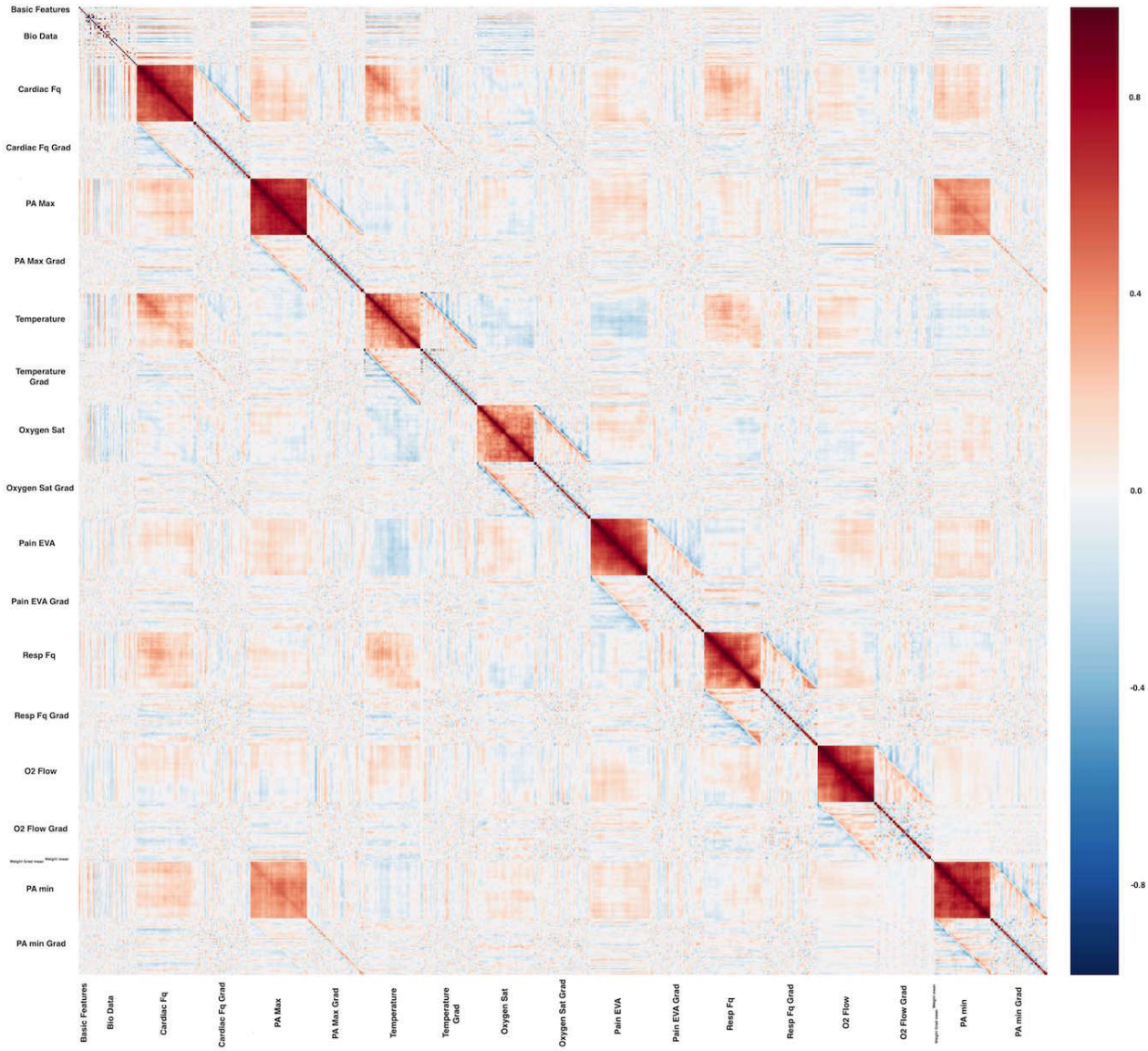

FIGURE 49: Matrice de corrélation linéaire sur l'ensemble des données avant le préprocessing

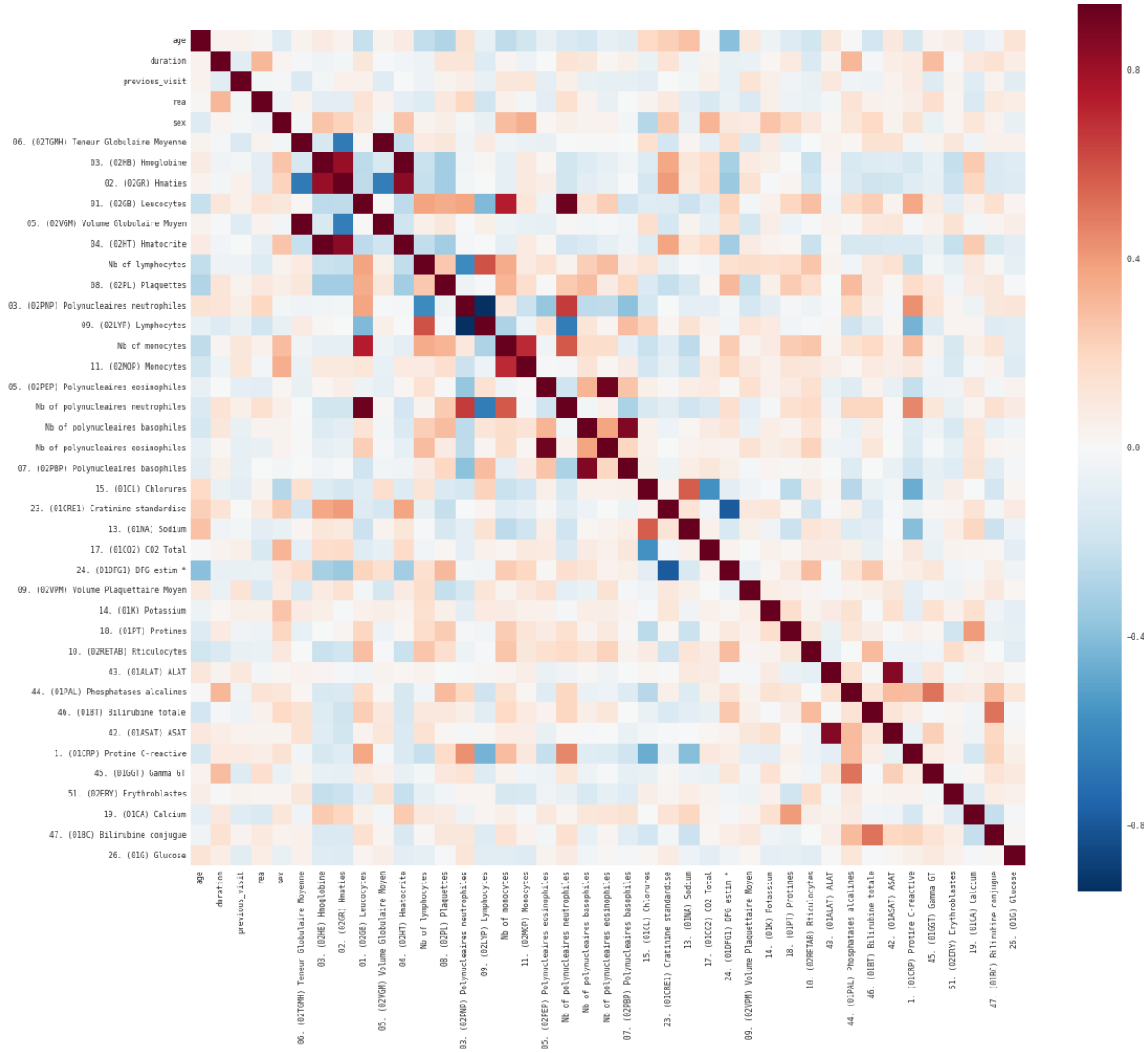

FIGURE 50: Matrice de corrélation linéaire sur les données biologiques avant le préprocessing

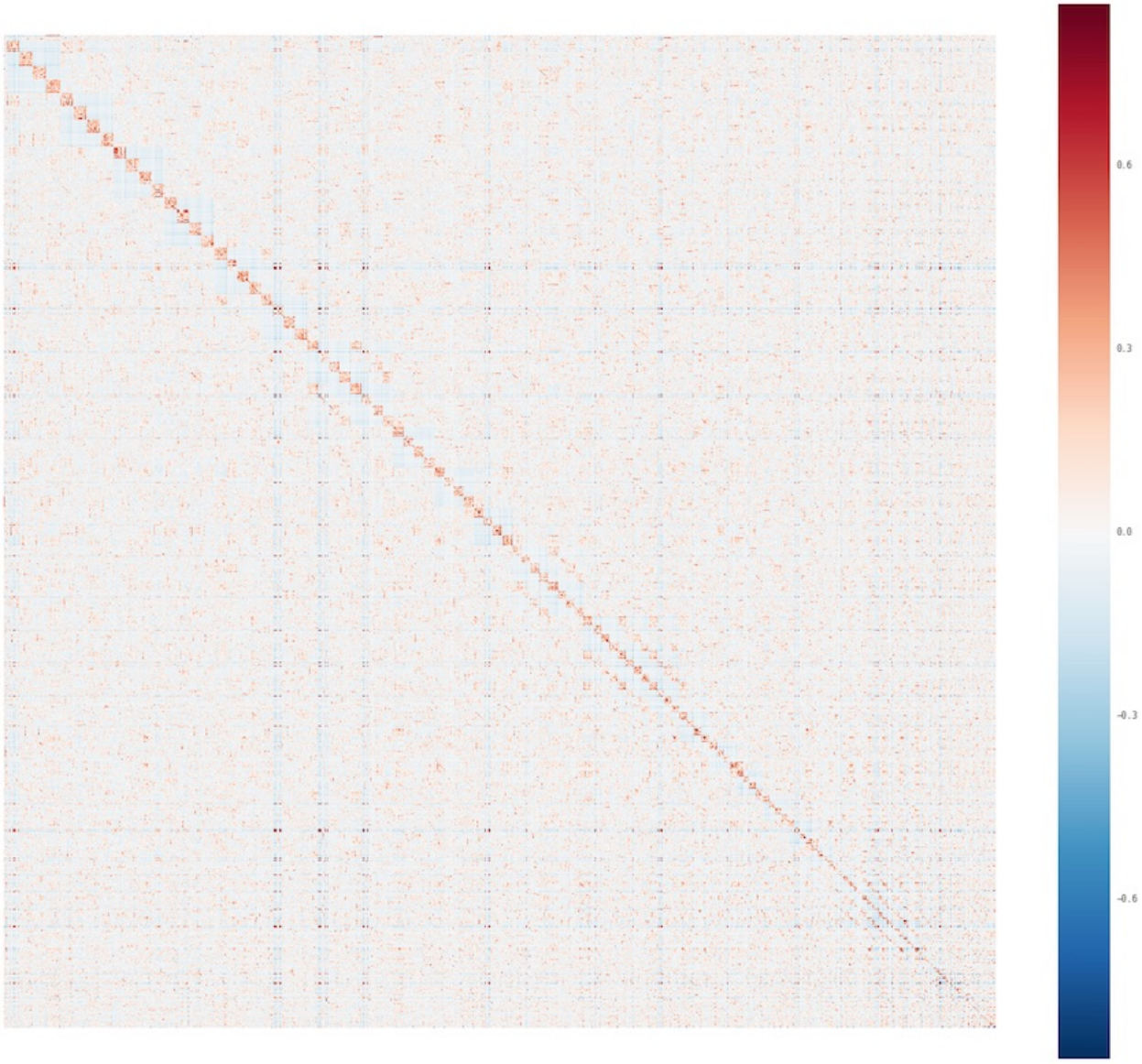

FIGURE 51: Matrice finale de corrélation linéaire sur l'ensemble des données de sortie (10546×10546)

Références

[1] Narayanaswamy Balakrishnan and Suvra Pal. Expectation maximization-based likelihood inference for flexible cure rate models with weibull lifetimes. *Statistical Methods in Medical Research*, 2013.

[2] SK Ballas and M Lusardi. Hospital readmission for adult acute sickle cell painful episodes : frequency, etiology, and prognostic significance. (pmid 15849770). *Am J Hematol. 2005 May*, pages 79(1) :17–25, 2005.

[3] Dries F Benoit, Rahim Alhamzawi, and Keming Yu. Bayesian lasso binary quantile regression. *Computational Statistics*, 28(6) :2861–2873, 2013.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[5] DC Brousseau, PL Owens, AL Mosso, JA Panepinto, and CA Steiner. Acute care utilization and rehospitalizations for sickle cell disease (pmid 20371788). *JAMA. 2010 Apr 7*, pages 303(13) :1288–94, 2010.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[7] Damien Garreau, Rémi Lajugie, Sylvain Arlot, and Francis R. Bach. Metric learning for temporal sequence alignment. *CoRR*, abs/1409.3136, 2014.

[8] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2) :83–85, 2005.

[9] Rein Houthooft, Joeri Ruyssinck, Joachim van der Herten, Sean Stijven, Ivo Couckuyt, Bram Gadeyne, Femke Ongenae, Kirsten Colpaert, Johan Decruyenaere, Tom Dhaene, et al. Predictive modelling of survival and length of stay in critically ill patients using sequential organ failure scores. *Artificial intelligence in medicine*, 63(3) :191–207, 2015.

[10] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[11] Harold C Sox, Marshall A Blatt, and Michael C Higgins. *Medical decision making*. ACP Press, 1988.

[12] Robert Tibshirani et al. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4) :385–395, 1997.

[13] Lian Wenzhao, Henao Ricardo, Rao Vinayak, Lucas Joseph, and Carin Lawrence. A multitask point process predictive model. *ICML*, pages 2030–2038, 2015.

[14] Eric Zapletal, Nicolas Rodon, Natalia Grabar, and Patrice Degoulet. Methodology of integration of a clinical data warehouse with a clinical information system : the hegp case. *Stud Health Technol Inform*, 160(Pt 1) :193–197, 2010.

[15] Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, pages 1567–1594, 2008.

[16] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778 : L-BFGS-B : Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4) :550–560, 1997.