ÉCOLE NATIONALE DE LA STATISTIQUE
ET DE L'ANALYSE DE L'INFORMATION

PROJET STATISTIQUE DE 2ÈME ANNÉE

PRÉDICTION CONFORME POUR LES DONNÉES TEXTUELLES ANALYSÉES PAR TRANSFORMERS

Etudiants :

Stuart BENOLIEL

Charles CARRERE

Louis THOMAS

Tuteur :

Stéphane CHRÉTIEN

Année universitaire 2023 - 2024

Abstract

This project centers on the application of conformal prediction in the healthcare domain, aiming to refine the predictive accuracy of patient outcomes using the MIMIC-III database. Conformal prediction, a key focus of this study, represents an innovative statistical approach that generates a set of possible outcomes with associated probabilities, significantly enhancing decision-making processes by providing a comprehensive probabilistic understanding of predicted outcomes. This methodology allows for the development of prediction intervals rather than singular predictions, offering a nuanced insight into the reliability of predictions made.

The research employs the BERT model as a tool for processing textual data contained within medical reports, transforming complex narratives into analyzable feature vectors. This step is crucial for extracting meaningful clinical information from unstructured data, facilitating the subsequent application of conformal prediction techniques. Following the BERT model, signatures are utilized to further distill and structure the feature vectors, enhancing the robustness and interpretability of the predictive models. The project specifically investigates the prediction of patient mortality and hospital stay duration, showcasing the potential of conformal prediction in offering actionable insights that could guide clinical decisions and resource allocation.

Utilizing the MIMIC-III database, the study meticulously applies these methodologies, demonstrating their viability and effectiveness in a clinical setting. Despite the promising results, the project acknowledges the inherent challenges posed by the complexity of medical data and the limitations of current methodologies, suggesting areas for future research and potential improvements.

Table des matières

1. Cette partie est largement reprise de la documentation MIMIC-III [8]

Table des figures

Introduction

Dans un contexte où la médecine cherche continuellement à affiner la précision de ses diagnostics et la personnalisation de ses traitements, l'analyse de données médicales volumineuses prend une importance capitale. Notre étude se propose d'utiliser la base de données MIMIC-III, afin de prédire les chances de survie des patients avant leur admission dans des études cliniques. S'inscrivant dans le cadre préparatoire à un projet ambitieux au Centre Léon Bérard de Lyon, qui s'intéressera à la survie ou non des patients en amount des essais cliniques. Ce travail s'attache à évaluer la portée de la prédiction conforme appliquée aux rapports médicaux. Cette technique prometteuse ouvre la voie à une assistance décisionnelle novatrice en proposant non seulement une prévision, mais aussi un ensemble de scénarios possibles, offrant ainsi aux médecins un outil précieux pour affiner leur jugement clinique et réduire le temps consacré à l'examen des rapports médicaux.

Notre recherche s'oriente vers la prédiction conforme, une approche statistique innovante qui, en dépassant les prédictions uniques, propose un ensemble de résultats possibles. Cela représente un avantage significatif pour les praticiens qui, submergés par la masse d'informations cliniques, ont besoin de synthétiser rapidement les données pour orienter efficacement leur jugement. La prédiction conforme s'avère donc être un atout précieux, agissant comme un filtre qui éclaire les décisions cliniques sans nécessiter une lecture exhaustive de chaque rapport.

Dans le cadre de notre étude, l'emploi de BERT (Bidirectional Encoder Representations from Transformers) est devenu un élément central. Ce modèle de traitement du langage naturel, développé par Google, est remarquable par sa capacité à comprendre le contexte des mots dans un texte. Cela le rend particulièrement adapté à l'analyse des rapports médicaux, où la précision sémantique est essentielle. BERT nous a permis de transformer des données textuelles complexes en vecteurs de caractéristiques exploitables, qui capturent non seulement le sens littéral des termes médicaux, mais aussi les nuances subtiles et les implications cliniques souvent présentes dans les notes des praticiens. L'utilisation de signatures constitue une autre dimension innovante de notre étude. Issue de la théorie des chemins rugueux en mathématiques, cette technique a le potentiel de capter l'essence dynamique des séquences de données cliniques dans le temps. En transformant des séries temporelles complexes en représentations succinctes mais informatives, les signatures saisissent les tendances et les motifs sous-jacents dans les trajectoires des paramètres physiologiques des patients.

Notre exploration débute par un cadre méthodologique approfondi. Une focalisation particulière est accordée à l'usage de la base de données MIMIC-III, discutant de son exploitation méthodique pour soutenir nos analyses. Nous présentons ensuite le détail des trois méthodes statistiques que nous utilisons, à savoir BERT, les signatures et en particulier les différentes approches de la prédiction conforme.

Nous progressons ensuite vers des applications spécifiques de ces méthodes, visant d'abord la prédiction de la survenue de la mort, avant de nous intéresser à l'évaluation de la durée de séjour hospitalière. Chacune de ces sections détaille la mise en œuvre pratique de nos approches, l'application des modèles prédictifs et l'interprétation des résultats obtenus.

1 Cadre méthodologique

1.1 Données utilisées : MIMIC III

La base de données MIMIC-III (Medical Information Mart for Intensive Care III) est une ressource clé pour les chercheurs en santé, fournissant des données anonymisées et détaillées sur les patients des unités de soins intensifs. Élaborée par le MIT Lab for Computational Physiology et le Beth Israel Deaconess Medical Center à Boston, elle vise à faciliter la recherche en santé digitale et l'analyse de données.

MIMIC-III compile les données de plus de 40 000 patients ayant été admis dans des unités de soins intensifs entre 2001 et 2012. Elle contient une variété d'informations, incluant les caractéristiques démographiques des patients, les mesures physiologiques, les résultats de laboratoire, les notes cliniques, les données sur les médicaments, les diagnostics, et les procédures. Cette richesse d'informations permet aux chercheurs de mener des études complexes et d'élaborer des modèles prédictifs.

Pour accéder à MIMIC-III, les chercheurs doivent suivre une formation sur la protection des données des patients et obtenir une autorisation, assurant la conformité avec les normes éthiques.

1.1.1 Méthodes [1]

La base de données MIMIC-III a été constituée à partir de données collectées durant les soins habituels à l'hôpital, ce qui n'a pas ajouté de charge de travail pour le personnel soignant ni perturbé leur routine quotidienne. Les informations proviennent de différentes sources, notamment :

— les systèmes informatiques des unités de soins intensifs ;
— les dossiers médicaux électroniques de l'hôpital ;
— le fichier des décès de l'Administration de la Sécurité Sociale.

Durant la période de collecte des données, deux systèmes d'information dédiés aux soins intensifs ont été employés : Philips CareVue Clinical Information System (modèles M2331A et M1215A ; Philips Healthcare, Andover, MA) et iMDsoft MetaVision ICU (iMDsoft, Needham, MA). Ces systèmes ont fourni des données cliniques précieuses, telles que :

— les mesures physiologiques contrôlées par les infirmières et enregistrées avec la date et l'heure, comme le rythme cardiaque, la pression artérielle ou la fréquence respiratoire, documentées chaque heure ;
— les comptes rendus de suivi rédigés par les soignants ;
— les détails sur les médicaments administrés en perfusion intraveineuse et le bilan des liquides.

Avant d'être intégrées dans la base de données MIMIC-III, les données ont d'abord été anonymisées conformément aux normes de la Health Insurance Portability and Accountability Act (HIPAA) en utilisant un nettoyage structuré des données et un décalage des dates. Le processus d'anonymisation pour les données structurées nécessitait la suppression des dix-huit éléments d'identification énumérés dans la HIPAA, y compris des champs tels que le nom du patient, le numéro de téléphone, l'adresse et les dates. En particulier, les dates ont été décalées vers le futur par un décalage aléatoire

1. Cette partie est largement reprise de la documentation MIMIC-III [8]

pour chaque patient de manière cohérente afin de préserver les intervalles, résultant en des séjours qui ont lieu quelque part entre les années 2100 et 2200. L'heure du jour, le jour de la semaine et la saisonnalité approximative ont été conservés lors du décalage des dates.

Les informations de santé protégées ont été retirées des champs de texte libre, tels que les rapports de diagnostic et les notes des médecins, en utilisant un système d'anonymisation rigoureusement évalué basé sur des recherches dans un dictionnaire étendu et une correspondance de motifs avec des expressions régulières. Les composants de ce système d'anonymisation sont continuellement étendus à mesure que de nouvelles données sont acquises.

1.1.2 Utilisation

Dans le cadre de ce projet, nous utilisons deux tables de cette base de données : `ADMISSIONS` qui rassemble les données concernant les hospitalisations uniques et `NOTEEVENTS` qui comprend les rapports médicaux collectés à l'issue d'une hospitalisation.

Admissions

Nom en Français	**Type de Donnée**
Identifiant de Ligne	Entier
Identifiant du Patient	Entier
Identifiant de l'Hospitalisation	Entier
Date et Heure d'Admission	Date et Heure
Date et Heure de Sortie	Date et Heure
Date et Heure de Décès	Date et Heure
Type d'Admission	Texte court
Lieu d'Admission	Texte court
Lieu de Sortie	Texte court
Type d'Assurance	Texte long
Langue du Patient	Texte très court
Religion du Patient	Texte court
État Civil du Patient	Texte court
Ethnicité du Patient	Texte moyen
Date et Heure d'Enregistrement aux Urgences	Date et Heure
Date et Heure de Sortie des Urgences	Date et Heire
Diagnostic à la Sortie	Texte long
Indicateur de Décès à l'Hôpital	Booléen
Données d'Événements Cliniques	Booléen

Nombre d'observations : 58976

TABLE 1 – Table des variables de la table `ADMISSIONS`

Dans notre étude, nous exploitons l'identifiant du patient ainsi que les dates d'admission et de sortie, ou la date du décès, afin de déterminer la durée du séjour hospitalier (en heures). Cette dernière constitue la variable principale de notre intérêt, en association avec l'indicateur de décès survenu à l'hôpital.

Les données proviennent de la base de données de l'hôpital relative aux admissions, aux sorties et aux transferts. Il arrive que des comptes de donneurs d'organes soient créés pour des patients décédés à l'hôpital. Ces comptes correspondent à des admissions hospitalières distinctes caractérisées par des durées de séjour très courtes, parfois même négatives. Pour traiter cela, nous ne prenons pas en compte les lignes avec une durée de séjour négative ou nulle.

Rapport Médicaux

Nom en Français	**Type de Donnée**
Identifiant de Ligne	Entier
Identifiant du Patient	Entier
Identifiant de l'Hospitalisation	Entier
Date du Compte Rendu	Date et Heure
Heure du Compte Rendu	Date et Heure
Heure de Saisie	Date et Heure
Catégorie de la Note	Texte court
Description de la Note	Texte long
Identifiant du Groupe Clinique	Entier
Indicateur d'Erreur	Caractère
Texte de la Note	Texte

TABLE 2 – Table des variables de la table `NOTEEVENTS`

Dans le cadre de notre analyse, nous nous concentrons sur le contenu textuel des notes cliniques, extraites de la table `NOTEEVENTS`. Ces textes représentent notre principal sujet d'étude et nous permettent d'extraire des informations cliniques significatives, telles que les symptômes, les diagnostics, les interventions et les résultats des traitements, qui sont essentiels pour notre recherche.

1.1.3 Traitement des données

Dans le contexte de cette étude, une mise au point concernant le traitement de nos données s'avère indispensable. La base de données offre un accès aux admissions hospitalières d'individus. Chaque admission s'étend sur plusieurs jours, durant lesquels des rapports à intervalles irréguliers sont consignés dans un fichier distinct. Il est possible que plusieurs rapports soient renseignés le même jour pour un même patient. Notre projet se concentre sur deux aspects principaux : la survie du patient pendant son hospitalisation et la durée de celle-ci en heures. Ainsi, notre unité statistique est représentée par une admission plutôt que par un patient, ce qui signifie qu'un même patient peut être pris en compte plusieurs fois s'il a eu plusieurs admissions. Cependant, dans le cadre de notre analyse, nous présumons une interchangeabilité des données, c'est-à-dire que l'ordre des observations n'a pas d'importance. Par conséquent, pour les patients ayant eu plusieurs séjours à l'hôpital dont un se concluant par le décès, seules l'admission aboutissant au décès est considérée dans notre étude. Nous supposons également que cette interchangeabilité des données demeure cohérente pour les patients ayant eu plusieurs admissions et ayant toujours survécu à celles-ci.

Nous avons exclu les donneurs d'organes, qui font l'objet d'un formalisme spécifique dans les données, ainsi que les individus dont la cohérence entre la date d'admission et la date de sortie était inadéquate. De plus, nous avons restreint notre analyse aux individus possédant au moins un rapport

médical lié à leur admission. Suite à ces prétraitements, notre ensemble de données exploitable comprend 58253 admissions provenant de 46081 individus distincts. Parmi ces 58253 admissions, 5550 se sont soldées par un décès du patient et 52703 par la survie du patient.

Pour entraîner par la suite nos modèles prédictifs, nous avons constitué un ensemble d'apprentissage comportant $n_{train} = 5000$ admissions, nombre déterminé empiriquement à l'aide de la Figure 12 en annexe [1], comprenant un nombre égal de cas de survie et de décès. L'ensemble de test totalise $n_{test+calibration} = 2000$ observations, dont 200 cas de décès et 1800 de survie, afin de maintenir des proportions similaires à celles de l'ensemble originel. Parfois, l'échantillon test est réduit pour former un ensemble de calibration, que nous évoquerons plus tardivement dans le cadre de la prédiction conforme.

Pour chaque admission sélectionnée dans l'un des échantillons, nous récupérons tous les rapports associés ainsi que les dates d'émission de ces rapports. Cette dimension temporelle sera exploitée dans le cadre de la théorie des signatures. Étant donné que l'objet TIMESTAMP n'est pas exploitable, nous représentons la dimension temporelle des rapports de la manière suivante : le premier rapport assigné à une admission est défini comme le jour 0, et les dates des autres rapports sont converties en nombres entiers de jours écoulés par rapport au jour 0. Comme plusieurs rapports peuvent être émis le même jour pour une même admission, nous avons rajoutés un terme de bruit aléatoire positif (tiré selon une loi uniforme entre 0 et 10^{-2}) sur les jours concernés pour éviter les doublons. Les rapports ne sont pas simplement concaténés pour former un seul rapport afin de conserver cette dimension temporelle.

En ce qui concerne le traitement NLP, nous extrayons les embeddings à partir du token CLS en utilisant le modèle de langue pré-entraîné (Bio_ClinicalBERT) et son tokenizer correspondant. Le texte est découpé en token, puis les embeddings du token [CLS] pour chaque partie sont récupérés. En raison du découpage des rapports en plusieurs parties si celui-ci dépasse 512 tokens, la longueur maximale prise en charge par de nombreux modèles pré-entraînés comme BERT, la dimension temporelle est dupliquée selon le nombre de découpages du rapport.

En raison de la grande taille des embeddings, une réduction de dimension s'avère nécessaire. Nous avons donc appliqué une Analyse en Composantes Principales (ACP) avec 100 composantes aux embeddings. Avec 100 composantes, la variance totale expliquée est assez grande aux alentours des 90% pour les deux échantillons. Ensuite, nous avons ajouté à la fin de chaque embedding la dimension temporelle mentionnée précédemment. Les admissions se concluant sur un unique rapport inférieur à 512 tokens et donc un unique embedding de taille 100 sont exclues car ne pouvant être utilisé en terme de signature (nécessité d'au moins deux points pour définir un chemin).

Une fois les embeddings réduits, nous calculons les signatures. Ces signatures capturent des informations complexes dans les données, ce qui est précieux lors de l'entraînement d'un modèle de prédiction. Cela conduit à la création d'un dictionnaire où chaque patient est représenté par des embeddings sous forme de signatures, avec un ordre de signature tronquée de 2 pour réduire encore la dimension.

Après avoir effectué l'extraction, la réduction de dimension et le calcul des signatures pour les

1. Ce graphique a été réalisé au début du projet, avant plusieurs modfications en terme de traitement et de prédiction, nous considérons néanmoins que l'ordre de grandeur des 5000 observations reste adapté

embeddings, les résultats sont transformés en un DataFrame Pandas.

Ce processus a réduit la taille de nos échantillons à n_{train} = 4989 et $n_{test+calibration}$ = 1995. Le nombre de features créées à partir des rapports et de ces processus est de 10302, avec des valeurs flottantes positives ou négatives.

1.2 Méthodes utilisées

1.2.1 BERT

Réseau de neurones

Inspirés par la compréhension des réseaux biologiques du cerveau humain depuis les années 1940, les réseaux de neurones sont des modèles informatiques utilisés pour simuler la manière dont les neurones interagissent pour traiter l'information. Composés de couches de neurones, chaque réseau contient plusieurs nœuds ou unités qui reçoivent des entrées, effectuent des calculs simples et produisent des sorties. Les connexions entre ces nœuds comportent des poids ajustables qui sont modifiés pendant l'apprentissage, permettant au modèle de s'adapter et de s'améliorer sur des tâches spécifiques telles que la reconnaissance d'images ou la prédiction de données. Ce processus d'apprentissage permet au réseau de neurones d'effectuer des tâches sans programmation explicite, imitant ainsi une forme d'apprentissage automatique. Dans le cadre de ce projet, nous allons utiliser un modèle BERT qui est un réseau de neurones particulier.

Modèles de langage à grande échelle

Dans le domaine en constante évolution du traitement du langage naturel (NLP), l'arrivée des modèles de langage à grande échelle (LLM) marque une étape importante, révolutionnant la manière dont les machines comprennent et interagissent avec le langage humain. Ces architectures avancées, à l'instar de BERT (Bidirectional Encoder Representations from Transformers)[4], ont démontré une capacité sans précédent à saisir les nuances et les contextes linguistiques, ouvrant la voie à des applications innovantes en matière d'interaction homme-machine et de compréhension textuelle.

Le principal atout de ces modèles réside dans leur aptitude à traiter de vastes corpus textuels, extrayant des connaissances et discernant des motifs subtils inaccessibles aux approches antérieures. Cette capacité d'analyse approfondie s'est traduite par des améliorations notables dans diverses tâches de NLP, telles que la classification de texte, la traduction automatique, et l'extraction d'informations, renforçant ainsi les fondements pour des percées majeures dans des secteurs tels que les services d'assistance virtuelle, la modération de contenu, et l'analyse sémantique.

Les modèles de langage à grande échelle (LLM) s'appuient sur des architectures neurales avancées pour traiter et générer du langage naturel. Avant l'introduction de BERT et d'autres modèles basés sur les Transformers, des techniques telles que les réseaux de neurones récurrents (RNN) et les modèles d'attention étaient couramment utilisées. Ces architectures permettaient déjà de traiter des séquences de mots pour produire des représentations vectorielles du langage, mais elles présentaient

des limitations, notamment en termes de capacité à gérer les longues dépendances et à paralléliser les calculs.

Bidirectional Encoder Representations from Transformers (BERT)

L'introduction des Transformers a marqué un tournant, offrant une méthode plus efficace pour capturer les relations entre tous les mots d'une phrase, indépendamment de leur position relative. Cette architecture repose sur le mécanisme d'attention, qui pondère l'importance relative de différents mots pour la compréhension d'un mot donné.

BERT (Bidirectional Encoder Representations from Transformers) représente une avancée majeure dans le domaine du TLP, en s'appuyant sur une structure de Transformer entièrement bidirectionnelle. Contrairement aux approches antérieures, qui traitaient le texte de manière séquentielle, BERT analyse simultanément le contexte à gauche et à droite de chaque token, permettant ainsi une compréhension globale du texte. Cette capacité d'encodage bidirectionnel est réalisée grâce à l'introduction d'un objectif de Modèle de Langage Masqué (MLM) lors du pré-entraînement.

Pré-entrainement de BERT

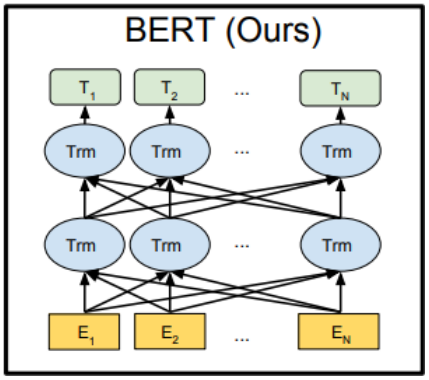


FIGURE 1 – Illustration de la structure de BERT

L'un des principaux innovations de BERT réside dans son objectif de Modèle de Langage Masqué (MLM). Contrairement aux approches traditionnelles de modélisation du langage qui prédisent chaque mot suivant dans une séquence en se basant uniquement sur les mots précédents, le MLM de BERT permet une compréhension bidirectionnelle du contexte. Durant le pré-entraînement, un certain pourcentage de tokens d'entrée est aléatoirement masqué, et le modèle est entraîné pour prédire ces tokens masqués en se basant sur leur contexte non masqué. Cette méthode permet à BERT d'intégrer de manière équilibrée le contexte situé à la fois avant et après le token masqué, facilitant ainsi l'apprentissage de représentations riches et contextuellement nuancées.

En complément de l'objectif MLM, BERT intègre également une tâche de prédiction de la phrase suivante (Next Sentence Prediction, NSP), conçue pour améliorer la compréhension des relations entre les phrases. Pour ce faire, lors du pré-entraînement, le modèle reçoit des paires de phrases et doit déterminer si la seconde phrase suit logiquement la première dans le texte original. Cette tâche vise à doter BERT d'une meilleure aptitude à comprendre les structures et les enchaînements logiques

dans les textes, ce qui est particulièrement bénéfique pour des applications telles que les questions-réponses et la compréhension de textes.

Le pré-entraînement de BERT, à travers les objectifs MLM et NSP, constitue une rupture par rapport aux méthodes conventionnelles de modélisation du langage, en permettant une analyse contextuelle profonde et bidirectionnelle. Cette approche novatrice non seulement enrichit la représentation sémantique des mots mais prépare également le terrain pour un ajustement fin efficace sur diverses tâches de TALN, sans nécessiter de modifications substantielles de l'architecture du modèle.

Fine-Tuning de BERT

Le processus de fine-tuning (ajustement fin) constitue la touche finale de l'entraînement de BERT pour des tâches spécifiques. Durant cette phase, BERT est ajusté sur un ensemble de données spécifique à une tâche, permettant au modèle de spécialiser ses connaissances linguistiques générales à cette tâche particulière. Le fine-tuning implique la modification légère des poids du réseau à travers toutes les couches, en utilisant un taux d'apprentissage très faible. Ce processus bénéficie grandement de la riche compréhension contextuelle développée durant le pré-entraînement, facilitant l'atteinte de performances élevées avec relativement peu de données d'entraînement spécifiques à la tâche.

Pour optimiser le processus de fine-tuning, diverses stratégies sont employées. Par exemple, le choix du taux d'apprentissage, du nombre d'itérations, et de la taille du batch joue un rôle crucial dans l'efficacité de l'ajustement. De plus, pour certaines tâches, il peut être bénéfique d'introduire des couches supplémentaires spécifiques à la tâche au-dessus de l'architecture de BERT, bien que la puissance de BERT réside dans sa capacité à s'adapter à de nombreuses tâches avec peu ou pas de modification architecturale.

Le fine-tuning de BERT a démontré son efficacité sur une gamme étendue de tâches de TALN, allant de la compréhension de texte (par exemple, SQuAD pour la réponse aux questions) à la classification de texte (comme GLUE pour l'analyse de sentiments et la reconnaissance d'entité nommée). Dans chacun de ces domaines, BERT, grâce à son ajustement fin, a souvent atteint ou dépassé l'état de l'art, illustrant la polyvalence et la puissance de son architecture.

Clinical BERT

L'adaptation et l'évolution de BERT dans le domaine clinique représentent une avancée significative[6], soulignant la flexibilité et la capacité d'adaptation de ce modèle à des contextes spécialisés. Cette spécialisation a pour but de tirer parti des connaissances préalablement acquises par les modèles de base et de les affiner pour mieux comprendre et traiter le langage spécifique au domaine clinique. La démarche consiste à adapter les représentations linguistiques générales aux subtilités et aux exigences des textes cliniques, offrant ainsi une base solide pour l'analyse précise et la classification de ces textes. Les Clinical BERT Embeddings sont le fruit d'un entraînement supplémentaire (ou fine-tuning) de BERT sur des ensembles de données cliniques vastes et diversifiés, tels que les notes de sortie et les rapports de diagnostic issus de la base de données MIMIC-III.

Pour chaque tâche spécifique au domaine clinique, ces modèles BERT ont été affinés afin d'opti-

miser leurs performances sur des tâches telles que la reconnaissance d'entités nommées (NER) et la désidentification (de-ID), ainsi que sur la tâche d'inférence de langage médical (MedNLI). Le processus de fine-tuning a permis de spécialiser davantage les modèles pour ces tâches, en passant l'embedding de sortie de BERT à travers une unique couche linéaire pour la classification. Cette approche, bien que représentant une capacité modélisatrice inférieure en comparaison avec des architectures plus complexes telles que Bi-LSTM, est alignée sur l'objectif de démontrer l'efficacité des embeddings spécifiques au domaine clinique.

1.2.2 Signatures

La théorie des chemins rugueux[2], initialement introduite par Terry Lyons, représente une avancée significative dans l'étude des systèmes dynamiques et des séries temporelles. Au cœur de cette théorie se trouve la transformation de signature, un outil mathématique puissant pour analyser et caractériser les chemins dans un espace multidimensionnel. Cette transformation, parfois comparée à la transformation de Fourier pour sa capacité à encoder des informations complexes, diffère cependant fondamentalement par sa nature non linéaire et sa capacité à capturer des informations d'ordre et d'interaction entre les différentes composantes d'un chemin.

Transformation de signature

La transformation de signature est un concept central dans l'étude des chemins rugueux, offrant une méthode pour encoder des informations contenues dans des chemins ou des séries temporelles multidimensionnelles. À la différence de la transformation de Fourier qui décompose un signal en fréquences constitutives, la transformation de signature décompose un chemin en un ensemble de caractéristiques qui capturent les interactions dynamiques entre ses composantes au fil du temps. Cependant, contrairement à la transformation de Fourier qui traite chaque canal de manière indépendante et est intrinsèquement linéaire, la transformation de signature est profondément non linéaire et tient compte explicitement des combinaisons de canaux, permettant ainsi une capture riche et intégrée des informations d'ordre et de structure.

Utilisation des signatures

Tout d'abord, nous considérons chaque embedding produit par le modèle BERT comme un point dans un espace multidimensionnel. Chaque point est associé à un instant temporel, donnant au chemin formé par ces embeddings une structure temporelle. Nous appliquons ensuite la transformation signature ou log-signature à ce chemin d'embeddings.

Dans la formule de la transformation signature donnée ci-dessus, les t_i représentent les instants temporels auxquels les embeddings sont calculés, et $f(t_i)$ désigne l'embedding correspondant au rapport i. Ainsi, la formule peut être exprimée de la manière suivante :

$$SigK(f) = \left(\int \cdots \int_{0<t_1<\cdots<t_k<1} \frac{df}{dt}(t_1) \times \cdots \times \frac{df}{dt}(t_k)\, dt_1 \ldots dt_k \right)_{1 \leq k \leq K} \tag{1}$$

Cette formule calcule les intégrales itérées des dérivées des embeddings le long du chemin, capturant ainsi les interactions complexes et l'ordre des données au fil du temps.

Signatory et son utilisation

Signatory est une bibliothèque conçue pour faciliter les calculs liés à la transformation de signature et de logsignature, en se concentrant spécifiquement sur les applications en machine learning. Cette bibliothèque permet d'exécuter des calculs non seulement sur les processeurs centraux (CPU) mais également sur les unités de traitement graphique (GPU), offrant ainsi des performances améliorées et une intégration transparente avec des environnements d'apprentissage profond comme PyTorch. Signatory est reconnue pour être la première bibliothèque capable d'effectuer ces opérations sur GPU, marquant un progrès significatif dans le domaine.

Les contributions de Signatory ne se limitent pas à son support GPU. La bibliothèque introduit également plusieurs améliorations algorithmiques et stratégies de précalcul efficaces qui accélèrent de manière substantielle les calculs de signature et de logsignature, même sur CPU sans parallélisme. Ces améliorations incluent l'optimisation des opérations de multiplication et d'exponentiation, ainsi que des stratégies de précalcul qui permettent des requêtes efficaces sur des intervalles de données arbitraires.

Nous avons intégré la bibliothèque Signatory pour exploiter sa rapidité et sa capacité à traiter efficacement des séries temporelles complexes, telles que les rapports médicaux. Ces documents, caractérisés par des intervalles de temps irréguliers entre les observations, présentaient un défi analytique en raison de leur structure séquentielle non uniforme. Grâce à Signatory, nous avons réussi à extraire des caractéristiques significatives et en capturant l'évolution des états de santé des patients avec précision. L'utilisation de Signatory dans notre projet nous a permis non seulement d'améliorer le temps de traitement des données, grâce à ses capacités de calcul sur GPU, mais aussi d'approfondir notre compréhension des dynamiques complexes contenues dans les rapports médicaux.

Signature vs Log-signature

La différence principale pour nous entre les log-signatures [2] et les signatures est que la première génère une quantité de données plus modeste : nous nous retrouvions avec 5151 caractéristiques contre 10302 pour les signatures.

Dans un premier temps, nous avons appliqué les log-signatures aux embeddings. Cependant, en raison de résultats plus prometteurs obtenus récemment avec les signatures, nous avons opté pour cette méthode. Cela nous a néanmoins permis d'obtenir des résultats légèrement similaires.

1.2.3 Prédiction conforme

Il existe différentes méthodes de prédiction conforme que nous avons tenté d'appliquer dans le cadre de notre étude. La méthode dite **Full Conformal**, qui est statistiquement intéressante car elle ne nécessite pas d'échantillon de calibration (et donc pas de perte de données d'entraînement), est

2. Pour plus de précision sur les log-signatures, se référer à l'encadré du Tableau 6 en annexe

computationnellement lourde à appliquer car elle repose sur un entraînement du modèle de ML sur toutes les valeurs possibles de la variable d'intérêt (Section Annexe). Cette méthode est donc inutilisable dans le cas d'un problème de régression et assez lourde en temps de calcul dans le cas de classification multiples.

Une autre méthode ayant des propriétés de couverture assez intéressantes est le **Jacknife+**. Mais, du fait de la taille de nos données et donc de temps d'entraînement de nos modèles assez long, nous n'avons pu appliquer cette méthode ici. Toutefois, nous avons appliqué la méthode **CV+** (Cross validation) à K folder que nous détaillerons dans un encadré (Section 3.3). Nous signalons que cette méthode est équivalente à celle du **Jacknife+** pour un $K = n_{entraînement}$.

L'approche la plus répandue est la méthode dite **Split** ou **Inductive** reposant sur un échantillon supplémentaire de données, et qui est celle que nous allons détaillé ici. Nous nous plaçons donc dans le cadre où nous disposons d'un n échantillon dit de calibration $\{(X_i, Y_i)\}_{i=1}^{n}$ et nous souhaitons prédire la valeur de Y_{n+1} étant donné X_{n+1}. Une hypothèse nécessaire sur notre jeu de données est que les échantillons $\{(X_i, Y_i)\}_{i=1}^{n+1}$ sont interchangeables −par exemple, être i.i.d.− d'une distribution jointe arbitraire P_{XY} avec le vecteurs des covariables $X \in \mathbb{R}^p$ et la variable d'intérêt $Y \in \mathbb{R}$. Le but est de construire un intervalle de prédiction $C(X_{n+1}) \subseteq \mathbb{R}$ ayant une propriété de couverture marignale sans faire d'hypothèse sur la distribution, qui contient avec une assez bonne certitude Y_{n+1}. Ainsi, étant donné le paramètre de non-couverture α, nous obtenons

$$\mathbb{P}\{Y_{n+1} \in C(X_{n+1})\} \geq 1 - \alpha$$

pour n'importe quelle distribution P_{XY} et n'importe quelle taille de n.

Pour construire cette intervalle de prédiction $C(X_{n+1})$, nous définissons un score de non conformité $s(X, Y)$ basé sur la notion heuristique de l'incertitude d'un modèle entraîné de ML. Un score élevé souligne que le modèle se trompe dans sa prédiction. L'intervalle de prédiction est construit de la manière suivante :

$$C(X_{n+1}) = \{y : s(X_{n+1}, y) \leq \hat{q}\} \tag{2}$$

où

$$\hat{q} = \text{quantile}\left(s(X_1, Y_1), \ldots, s(X_n, Y_n); \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right)$$

2 Prédiction de la survenue de la mort

2.1 Mise en oeuvre

Suite au traitement des rapports, nous nous retrouvons avec un jeu de données finalement utilisables dans le but de prédire une variable intéressante dans le but d'un suivi clinique : la survenue ou non de la mort du patient. Cette variable est très déséquilibrée en terme des occurences de chacune de ses valeurs possibles. En effet, 52703 individus ont survécu durant leur séjour à l'hôpital contre 5500 décès seulement. Dans l'objectif de prédire cette variable binaire, nous avons utilisé les deux algorithmes de machine learning (ML) suivant :

- Une régression logistique : De manière générale la régression logistique est utilisée comme un modèle de classification supervisée pour prédire une variable cible binaire à partir de variables indépendantes. Le modèle opère par la fonction logistique, ou sigmoïde, qui transforme un résultat linéaire $\beta_0 + \sum_{i=1}^{k} \beta_i x_i$ en une probabilité entre 0 et 1, interprétée comme la probabilité qu'un événement appartienne à une classe spécifique. L'entraînement du modèle implique la maximisation de la fonction de vraisemblance, ce qui optimise l'ajustement des prédictions du modèle aux données observées. Cette méthode est particulièrement appréciée pour sa capacité à fournir des probabilités conditionnelles pour les prédictions, offrant ainsi une mesure quantifiable de l'incertitude associée à chaque classification. Nous utilison la pénalisation L1. Cette pénalisation produit des modèles parcimonieux et peut ainsi être utilisée pour effectuer une sélection de variables. Ceci est donc parfaitement adapté à notre jeu de données qui contient plus de 10000 variables après traitement des rapports.

- Une forêt aléatoire (RF) : Les forêts aléatoires (Random Forests) constituent une méthode de classification et de régression qui repose sur l'agrégation de nombreux arbres de décision. Chaque arbre est construit à partir d'un sous-ensemble des données d'entraînement, sélectionné aléatoirement, ce qui introduit de la diversité dans les modèles générés. En outre, lors de la construction de chaque arbre, la sélection des variables à chaque division est également effectuée aléatoirement parmi un sous-ensemble des variables disponibles. Lors de la phase de prédiction, les forêts aléatoires combinent les résultats de tous les arbres pour produire une sortie unique : pour la classification, cela se fait généralement par un vote majoritaire, tandis que pour la régression, il s'agit de la moyenne des prédictions de tous les arbres. Cette méthode est largement appréciée pour sa haute précision, sa robustesse face au surajustement, particulièrement avec des données de haute dimensionnalité, et sa capacité à gérer à la fois des variables catégorielles et continues.

La sélection de diverses hyperparamètres (max_features représentant le nombre de covariables maximales pour chaque division dans le cadre de la forêt aléatoire / le paramètre Cs représentant l'inverse de la force de régularisation pour la régression logistique) a été effectués via la fonction GridSearchCV pour la forêt et directement via LogisticRegressionCV pour la seconde grâce au package sklearn. Le nombre de variables étant très élevés, l'ensemble du jeu de données n'a pu être utiliser. Les modèles ont été entrainé pour un $n_{entraînement} \approx 5000$ observations, un échantillon test de $n_{test} \approx 1350$ et d'un échantillon de calibration pour la prédiction conforme de $n_{calibration} \approx 650$. Comme évoqué plus haut, les modèles ont été entrainé de sorte que les poids de chacune des classes soient proportionnels à l'inverse de la fréquence des classes dans le jeu d'entraînement qui a été conçu (tout comme le jeu de test) de sorte à conserver le ratio Survivant/Décès du jeu de données initiale.

Les résultats des deux modèles sont présentés ci-dessous pour un score de seuil égal de 0.5 :

Modèles	**Taux de bien classés**	**Rappel**	**F1-score**
LR	49.32%	43.00%	14.54%
RF	38.90%	79.00%	20.59%

TABLE 3 – Comparaison des résultats des deux modèles

On constate très rapidement que les résultats de nos modèles ne sont pas de très bonnes qualités. Le taux de bien classés est moyen pour la régression logistique et mauvais pour la forêt aléatoire. La forêt aléatoire compense d'une certaine manière avec un grand pourcentage de rappel (représentant le pourcentage de positifs bien prédits). En terme de F1-score qui est une moyenne harmonique entre le rappel et la précision, les deux modèles ont un faible taux. Ces résultats sont d'autant plus visible si l'on s'intéresse aux courbes ROC des deux modèles. Les distributions des scores sont également disponibles en Annexe figures 13a et 13b qui montrent bien que les modèles n'arrivent pas à distinctement séparer les deux classes. Nous allons voir dans ce cas si la prédiction conforme peut être utile dans une situation où les modèles ne sont pas très performants.

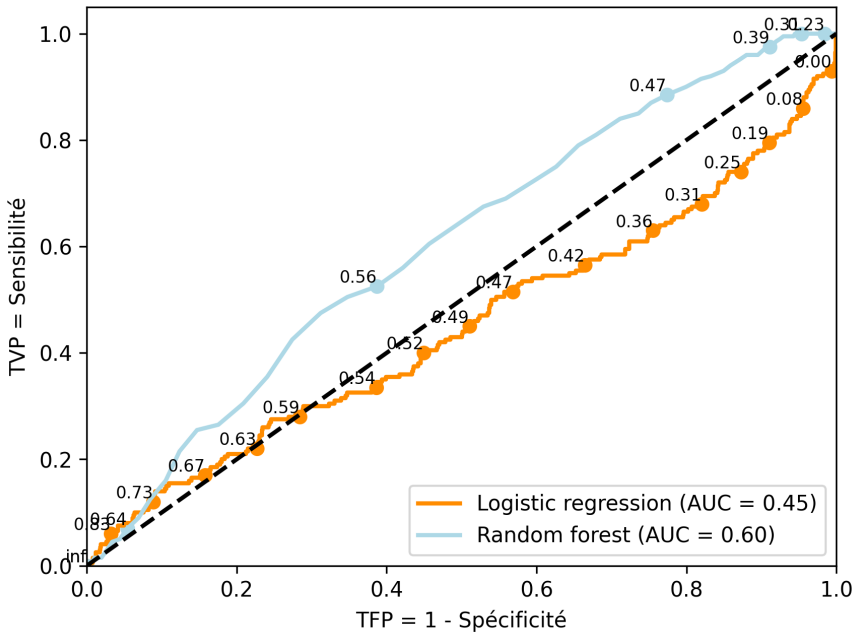
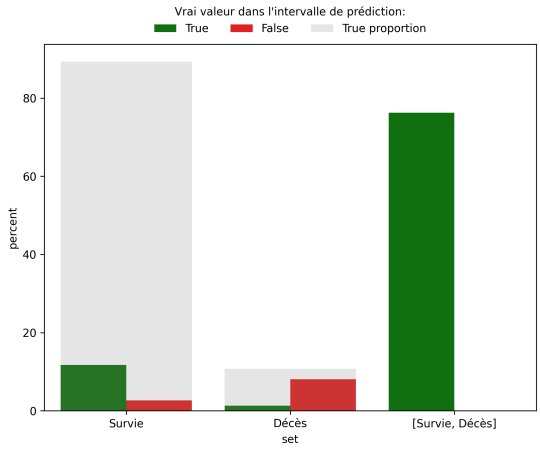


FIGURE 2 – Courbe ROC

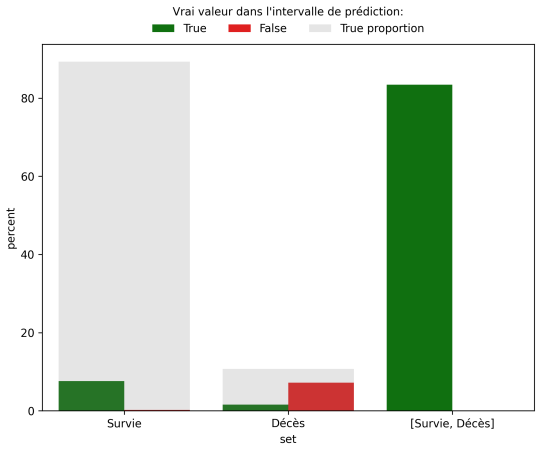
2.2 Applications de la prédiction conforme

L'approche classique de la prédiction conforme appliquée à un problème de classification est celle utilisée en préambule de l'article [1]. Comme évoqué en début de rapport, elle permet étant donné un paramètre de non-couverture α, d'obtenir des intervalles de prédictions ou plutôt des sets de valeurs contenant la vrai valeur pour approximativement $(1-\alpha)\%$ des cas. Et ceci quelque soit la qualité de notre classifieur $\hat{h}$ et sans faire d'hypothèses forte sur la distribution des données. Néanmoins, bien que la couverture soit effectivement atteinte en pratique, dans le cadre d'un problème de classification binaire, l'intérêt de la méthode se trouve fortement limitée et est même contre-productive.

Nous présentons l'application de la méthode évoquée comme produisant des sets de faibles tailles dans l'article et qui repose sur la fonction de score suivante $s(X_i, y) = 1 - \hat{h}(X_i)_y$ où $\hat{h}(X_i)_y$ correspond à la probabilité estimée d'appartenir à la classe y étant donné X_i.



(a) Régression logistique : couverture de 89.29%



(b) Forêt aléatoire : couverture de 92.57%

FIGURE 3 – Comparaison des différents sets de prédiction selon le modèle de ML

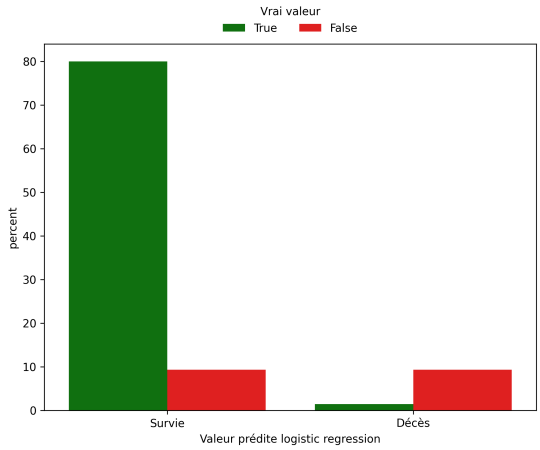
Nous remarquons très rapidement que dans environ 80% des cas, les sets contiennent les deux valeurs possibles. C'est un pourcentage bien trop élevé pour que ceci puisse être finalement exploité. En effet la non couverture d'environ 10% réside dans la prédiction des sets contenant uniquement une des deux valeurs ('Survie' ou 'Décès'). Ce qui veut dire que dans les 20% des prédictions finalement utilisables, nous nous trompons environ 1 fois sur 2. Peut-être que si nos classifieurs étaient plus performant, cette méthode aurait eu un plus grand intérêt.

Une autre méthode mentionée dans l'article [1] serait celle de l'"Outlier detection". Au lieu de prédire un set pouvant comprendre plusieurs valeurs à la fois, notre set devient de la forme suivante $C(X_{n+1}) \in \{\text{outlier}, \text{inlier}\}$ où outlier correspondrait au label 'Décès' tandis que inlier au label 'Survie'. Cette méthode garantie que $P(C(X_{n+1}) = \text{outlier}) \leq \alpha$ c'est à dire que nous prédisions 'Décès' dans moins d'α% des cas. L'intervalle de prédiction est construit de la manière suivante :

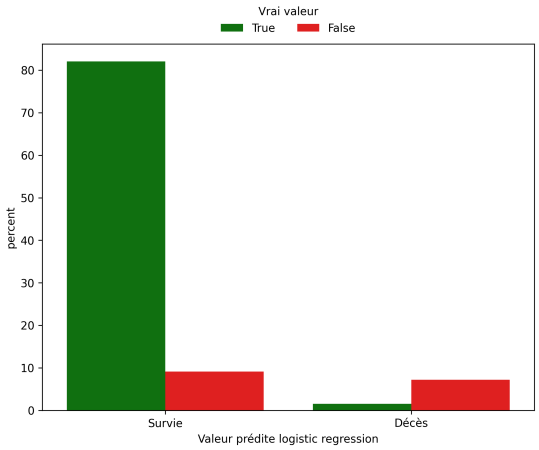
$$C(X_{n+1}) = \begin{cases} \text{inlier} & \text{si } s(X_{n+1}) \leq \hat{q} \\ \text{outlier} & \text{sinon} \end{cases} \tag{3}$$

où

$$\hat{q} = \text{quantile}\left(s(X_1), \ldots, s(X_n); \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right) \text{ et } s(X_i) = 1 - \hat{h}(X_i)$$



(a) Régression logistique : couverture de 81.41%



(b) Forêt aléatoire : couverture de 83.64%

FIGURE 4 – Comparaison des différentes prédictions via méthode outlier

Pour les deux classifieurs, la couverture globale est aux alentours des 80% ce qui est mieux qu'avec un score fixé à 0.5 en comparaison avec le tableau 3. Pour un jeu de données avec cette caractéristique d'outlier-inlier, on voit que cette méthode de prédiction conforme donne une façon de choisir la valeur limite pour le score. Notre prédiction des 'Décès' est bien inférieure ou à peu près égale à 10% comme convenu. Cependant, environ toutes les prédictions du label 'Décès' sont fausses ce qui veut dire que cela revient au même de prédire 'Survie' pour tous les individus. Nous rendons compte que la prédiction conforme ne permet pas de pallier la faiblesse de nos classifieurs, nous décidons donc d'opter une nouvelle approche.

3 Prédiction de la durée de séjour à l'hôpital

Les résultats peu fructueux de nos algorithmes de machine learning, appliqués à la variable binaire "Survie durant le séjour à l'hôpital", nous ont poussés à changer le cadre d'analyse de notre étude pour nous tourner vers un autre objectif. Au lieu de nous contenter d'une variable binaire où l'intérêt de la prédiction conforme se voit limité, nous nous sommes concentrés sur la prédiction de la durée de séjour des patients (Y_i). Ce choix, au lieu de prédire une durée de vie, a été fait pour éviter de se placer dans le cadre des modèles de survie, complexifiant d'autant plus l'étude et l'application souhaitée de la prédiction conforme.

3.1 Étude de la nouvelle variable durée de séjour

	Nombre d'observations	**Moyenne (H)**	**Médiane (H)**	**Variance**	**Min (H)**	**Max (H)**
Total	58253	244.34	156.7	299.5	0.03	7071.85
Survie	52703	243.80	158.27	295.16	0.35	7071.85
Décès	5550	249.5	144.40	338	0.03	4954.22

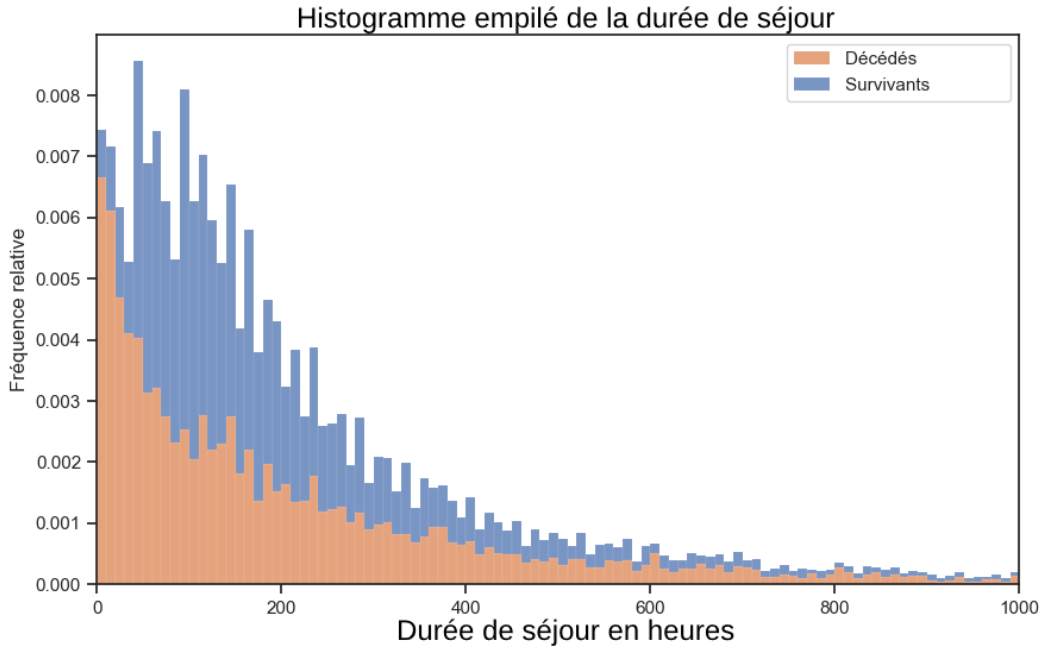


FIGURE 5 – Histogramme de la durée de séjour

Même si les moyennes des deux groupes semblent similaire, c'est en réalité due au fait que la distribution est très étalée à droite (cf Annexe figure 11). On peut voir sur l'histogramme que la distribution décroit strictement pour les décès alors qu'il y a un pic pour les survies. En étudiant l'histogramme on peut voir qu'il y a un pic autour de 50H pour la durée de séjour des survivants.

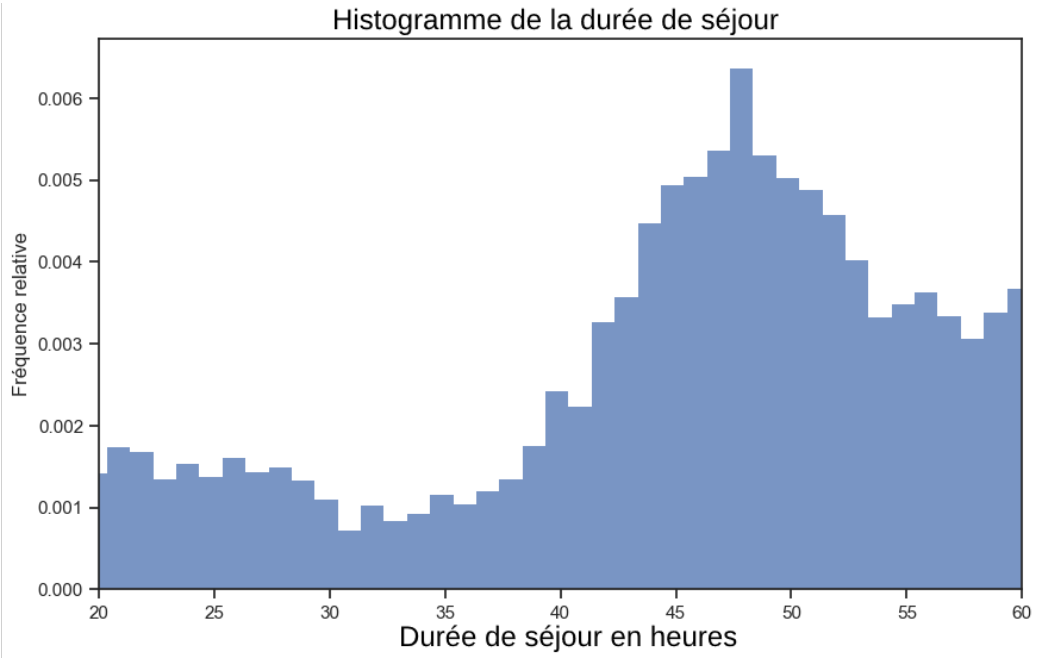


FIGURE 6 – Histogramme de la durée de séjour

On remarque que ce pic très important se situe autour de 48H. Nous n'avons pas d'information concernant la raison de ce pic mais étant donné la nature de la donnée, on peut penser que cela est du à une procédure de surveillance des patients durant 48H dans les hôpitaux.

3.2 Mise en oeuvre

La première approche est l'approche classique de la prédiction de l'espérance conditionnelle de Y sachant $X = x_i$ via forêt aléatoire. Le modèle a été entraîné sur le même échantillon de $n_{entraînement} \approx$ 5000 en minimisant l'erreur quadratique moyenne. L'hyperparamètre du nombre maximal de variables à considérer pour chaque coupure (max_features) a été optimisé via cross-validation. Comme nous entraînons une forêt, nous récupérons comme mesure d'erreur de notre prédicteur l'erreur Out-Of-Bag (OOB). Sur notre échantillon test $n_{test+calibration} \approx 2000$, nous regardons également l'erreur quadratique moyenne ainsi que le R^2.

OOB	MSE	R^2
0.34	57373	0.28

TABLE 4 – Résultat de la forêt aléatoire

L'erreur quadratique moyenne est de l'ordre de la valeur moyenne de la durée de séjour au carré bien qu'un peu plus élevée. Au vu des résultats obtenus, on se rend compte qu'après traitement des rapports médicaux, le pouvoir prédictif du modèle est non négligeable mais reste limité. C'est pourquoi nous nous sommes tournés vers d'autres approches.

Une approche moins conventionnelle mais qui semble particulièrement appropriée à notre objectif de prédire un intervalle de prédiction avec une couverture satisfaisante est d'utiliser une régression quantile.

Régression quantile

Le but de la régression quantile est d'estimer le quantile conditionnelle d'ordre alpha de la distribution :

$$Q_\alpha(x) := \inf\{y \in \mathbb{R} : P(Y \leq y | X = x) \geq \alpha\}$$

via le problème d'optimisation suivant :

$$\hat{Q}_\alpha(x) = f(x;\hat{\theta}), \quad \hat{\theta} = \text{argmin}_\theta \left[\sum_{i=1}^{n} \rho_\alpha(Y_i, f(X_i;\theta)) \right]$$

où $f(x;\theta)$ est la fonction de régression quantile et $\rho_\alpha$la fonction de perte quantile (Pinball loss)

$$\rho_\alpha(y,\hat{y}) := \begin{cases} (1-\alpha)(\hat{y}-y) & \text{si } \hat{y} \geq y \\ \alpha(y-\hat{y}) & \text{sinon} \end{cases}$$

Néanmoins ce n'est pas la méthode utilisée par la régression quantile par forêt aléatoire par exemple. Nous laissons le lecteur se réfèrer à l'article suivant pour plus de détail [7].

Dans l'objectif d'avoir une couverture d'ordre $1-\alpha$, on pourrait penser qu'il serait suffisant de prédire les quantiles $\hat{Q}_{\alpha/2}(x)$ et $\hat{Q}_{1-\alpha/2}(x)$ de sorte de définir $\hat{C}(X_{n+1}) = [\hat{Q}_{\alpha/2}(X_{n+1}), \hat{Q}_{1-\alpha/2}(X_{n+1})]$. Toutefois, la couverture n'est vérifiée qu'asymptotiquement. Sur notre échantillon test, la couverture

était surestimé (91.82%) au vu de l'objectif initial de 90%. Il est nécessaire d'effectuer une correction que nous effectuons comme une application de la méthode **Split Conformal** à la régression quantile. Nous choisissons d'utiliser une nouvelle fois une forêt aléatoire pour estimer ces quantiles en optimisant différents hyperparamètres via cross-validation.

3.3 Applications de différentes méthodes de prédiction conforme

Pour les deux approches, nous avons appliqué la méthode de prédiction conforme via **Split**.

Dans le cas de la moyenne conditionnelle de Y sachant $X = x_i$, nous appliquons la formule (2) avec la fonction de score définie par $s(X_i, y) = \left|y - \hat{f}(X_i)\right|$. Par ailleurs, comme la durée de séjour est forcément positive, nous corrigeons les intervalles de sorte qu'ils soient toujours inclus dans $\mathbb{R}^+$:

$$C(X_{n+1}) = [\max\{0, \hat{f}(X_{n+1}) - \hat{q}\}, \hat{f}(X_{n+1}) + \hat{q}] \tag{4}$$

où nous rappelons que

$$\hat{q} = \text{quantile}\left(s(X_1, Y_1), \ldots, s(X_n, Y_n); \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right)$$

Nous montrons ici par exemple, l'amplitude de l'intervalle ainsi que les valeurs prédites par notre régresseur $\hat{f}$ sur cent observations de l'échantillon test. La distribution de la taille des intervalles pour l'ensemble de l'échantillon test est également disponible en Annexe figure 15a.

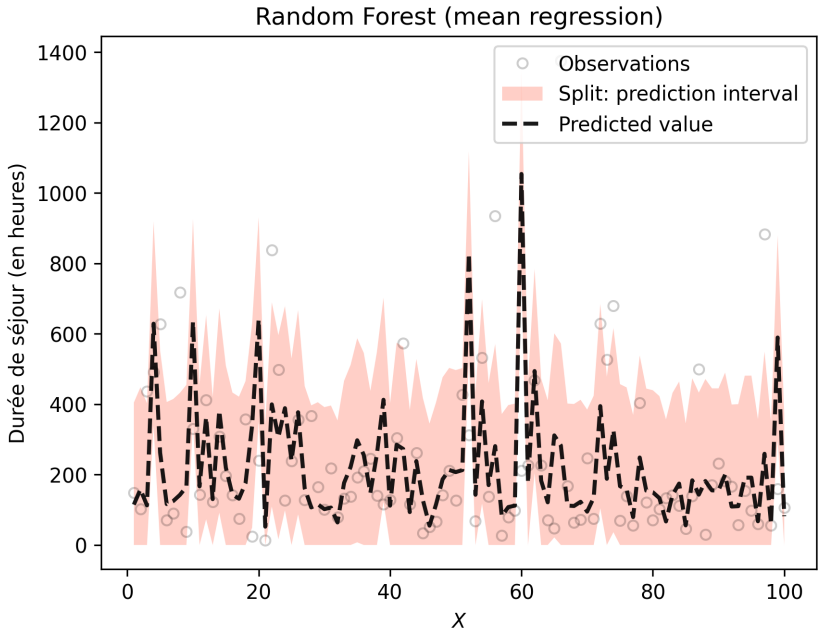


FIGURE 7 – Visualisation des intervalles de prédiction pour quelques observations de la régression moyenne via forêt aléatoire

Un autre méthode mentionnée précédemment dans notre article est la méthode **CV+** dont nous détaillons la pratique ci-dessous.

CV+ Validation croisée à K plis

Cette méthode ne nécessite pas d'échantillon de calibration contrairement à la méthode **Split**. Comme son nom le suggère, nous découpons notre échantillon d'entraînement de taille n noté S en K morceaux $S_1, S_2, ..., S_K$ de tailles approximativement égales à $\lfloor n_{entraînement}/K \rfloor$ (avec une correction si $K \nmid n$). Le modèle de machine learning est ensuite entraîné K fois en considérant l'échantillon d'entraînement privé de chacun des plis ($-S_k = S \setminus S_k$ où $k = 1, ..., K$) : $\hat{f}_{-S_1}, ..., \hat{f}_{-S_K}$. Nous définissons les résidus du processus de la manière suivante :

$$R_i = \left| Y_i - \hat{f}_{-S_{k(i)}}(X_i) \right|$$

avec $i = 1, ..., n$ et où $k(i) \in \{1, \ldots, K\}$ identifie le sous-ensemble qui contient i, c'est-à-dire $i \in S_{k(i)}$. Nous utilisons également la notation abusive :

$$\hat{q}\{v_i\} = \text{quantile}\left(v_1, \ldots, v_n; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right)$$

Et l'intervalle de prédiction :

$$C(X_{n+1}) = [-\hat{q}\{R_i - \hat{f}_{-S_{k(i)}}(X_{n+1})\}, \hat{q}\{R_i + \hat{f}_{-S_{k(i)}}(X_{n+1})\}]$$

Comme il n'est pas nécessaire de disposer un échantillon de calibration, notre échantillon d'entraînement utilisée pour l'entraînement du modèle de forêt aléatoire comprendra également les observations utilisées pour la calibration (et uniquement pour cette méthode). De plus, nous effectuons une nouvelle fois une correction de la borne inférieure de l'intervalle pour qu'il ne contienne aucune valeur négative. Du fait de la similitude des résultats obtenues avec la méthode précédente, nous laissons le lecteur se référer aux figures 14a et 15b situées en Annexe.

Pour l'approche utilisant la régression quantile, il est nécessaire d'utiliser une autre fonction de score du fait que nous prédisons non pas une mais deux valeurs :

$$s(X_i, y) = \max\{\hat{Q}_{\alpha/2}(X_i) - y, y - \hat{Q}_{1-\alpha/2}(X_i)\}$$

Nous pouvons ensuite utiliser l'intervalle définit par la formule (2). Nous ferons référence à cette méthode via le terme **CQR** (Conformalized quantile regression) pour éviter tout malentendu avec la première approche.

Nous montrons ici par exemple, l'amplitude de l'intervalle sur cent observations de l'échantillon test. La distribution de la taille des intervalles pour l'ensemble de l'échantillon test est également disponible en Annexe figure 15c.

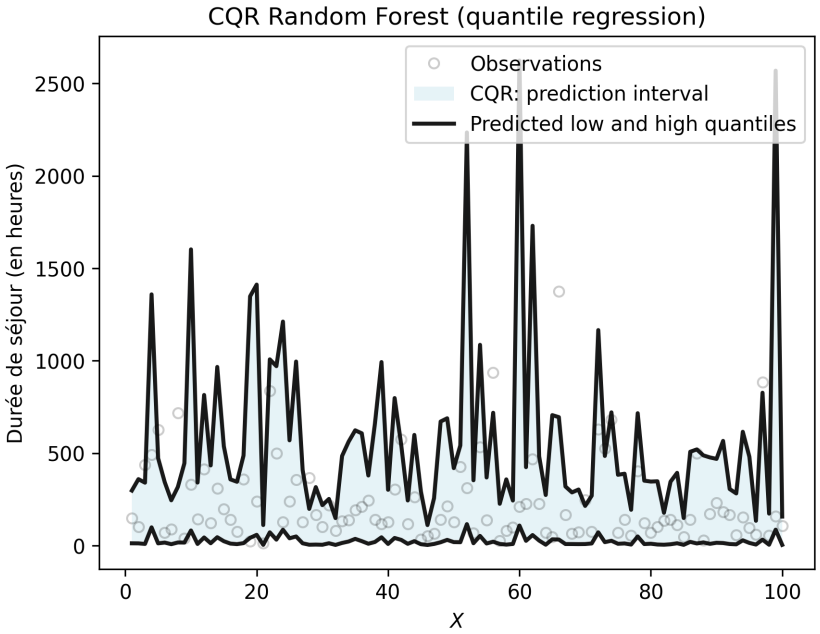


FIGURE 8 – Visualisation des intervalles de prédiction pour quelques observations de la régression quantile via forêt aléatoire

3.4 Analyse des résultats

Dans le cadre d'un problème de régression, plusieurs paramètres permettent d'évaluer les performances des différentes méthodes de prédiction conforme. Le pourcentage de bonne couverture sur l'échantillon test et la taille moyenne des intervalles de prédiction sont des indicateurs typiques pour mesurer finalement le fruit de nos prédictions. On s'intéresse également à l'écart-type de la taille de nos intervalles de prédictions. En effet, des intervalles de tailles assez dispersés sont plus intéressants que des intervalles de faible amplitude car ils présument de la non confiance de la prédiction de notre modèle sur la prédiction faite.

Pour comparer les résultats obtenues avec un cas optimal que nous nommerons **Best**, nous prenons les valeurs prédites de notre forêt aléatoire (pour une estimation de l'espérance conditionnelle de Y sachant $X = x_i$) à laquelle nous définissons un intervalle de prédiction centré et approximativement symétrique (nous faisons en sorte de garder des intervalles de valeurs positives en cohérence avec ce qui a été fait auparavant) à la valeur prédite :

$$\hat{C}(X_{n+1}) = [\max\{0, \hat{f}(X_{n+1}) - \frac{\left|Y_{n+1} - \hat{f}(X_{n+1})\right|}{2}\}, \hat{f}(X_{n+1}) + \frac{\left|Y_{n+1} - \hat{f}(X_{n+1})\right|}{2}]$$

La méthode **Best** permet de rendre compte de la manière dont fluctue notre taille d'intervalle de prédiction lorsque celle-ci est compliquée (la distance entre Y_{n+1} et $\hat{f}(X_{n+1})$ est grande) en comparaison avec les autres méthodes. Cette méthode a l'avantage de prendre en compte que $\hat{f}$ n'est pas parfait. Toutefois, la couverture est également parfaite alors qu'on s'autorise une non couverture de α%. Ce qui fait que pour les intervalles de prédiction très élevés, le modèle aurait pu se permettre de se tromper en réduisant leur taille. Ainsi pour un taux de couverture d'exactement $1 - \alpha$, la moyenne et

l'écart-type sont surestimés. Cependant, nous décidons de n'effectuer aucun traitement qui pourrait réduire artificiellement la couverture d'autant que cette méthode a juste pour but de donner une idée d'un cas optimal.

Le tableau suivant résume les différentes valeurs pour les méthodes appliquées :

Méthodes utilisés	**Taux de couverture**	**Moyenne**	**Médiane**	**Ecart-type**
RF Best	100.00%	242	148	276
RF Split	87.96%	470	461	77
RF CV+	90.48%	506	491	82
CQR	88.55%	527	413	467

TABLE 5 – Comparaison des différentes méthodes pour un $\alpha = 0.1$

D'une part, on voit que le taux de couverture de toutes les méthodes s'approchent bien les 90%. Comme c'est un paramètre que nous souhaitons atteindre et fixer, les méthodes **Split** et **CQR** font un peu plus de sous-couverture comparée à **CV+**. On voit de faibles écarts entre la méthode **Split** et **CV+** en terme d'indicateurs de dispersion centrales. On peut tout de même signaler et rappeler qu'au vu de l'expression (4), la longueur de l'intervalle de prédiction devrait être fixe et égale à $2\hat{q}$ (hors ajustement) pour la méthode **Split**. Ce qui fait que cette méthode est peu adaptative au sens où la longueur de l'intervalle ne dépend pas de l'aisance ou non du modèle à prédire la valeur. Dans le cas d'outlier, le modèle est enclin à se tromper et la prédiction conforme ne peut corriger cette aspect sans choisir des longueurs d'intervalles trop grand. On voit notamment sur la distribution des tailles d'intervalles que le modèle se trompe en grande partie dans la partie supérieure de la distribution (Annexe figure 15a) c'est à dire quand l'intervalle n'est pas tronqué et est bien égale à $2\hat{q}$. Tandis que la méthode **CV+** est censé être un peu plus adaptative. On le voit également sur la distribution qui est un peu plus étalée (Annexe figure 15b). Mais c'est particulièrement la méthode **CQR** qui sort du lot. On voit notamment un écart-type élevée ce qui est particulièrement appréciée au sens d'une meilleure adaptativité de la méthode. La moyenne est donc un peu plus élevée mais on voit bien qu'en terme de médiane elle surpasse ses consoeurs. Tout comme les erreurs de prédiction qui ne sont pas concentrées en queue de distribution.

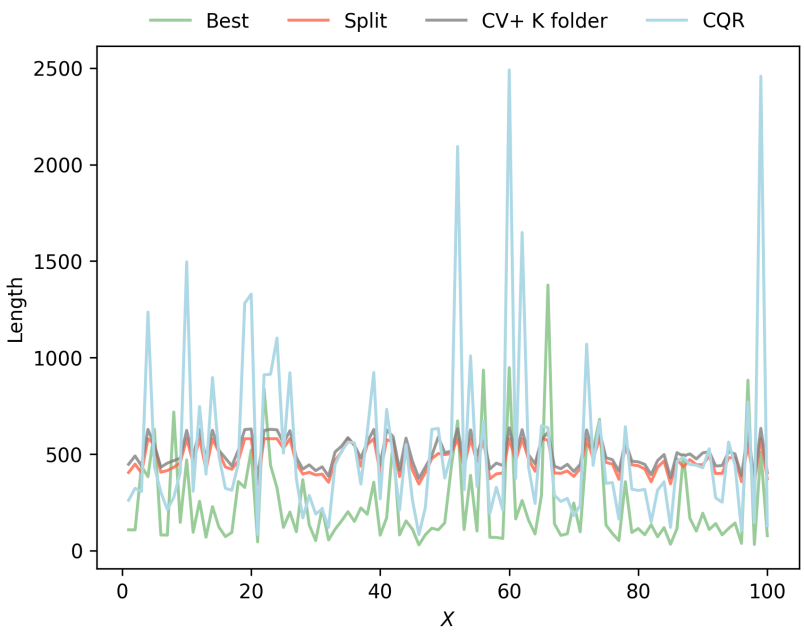


FIGURE 9 – Evolution de la taille des intervalles obtenues pour les différentes méthodes

On peut également voir comment évolue la taille des intervalles sur cent mêmes observations pour comprendre leur mécanisme. On voit que les courbes associés à la **Split** et au **CV+** sont quasi-identiques et peu adaptatives. En comparant la méthode **CQR** et la **Best**, on voit qu'elles suivent toutes les deux la même tendance. La méthode **CQR** est plus volatile et fournit des intervalles toujours un peu plus grand que nécessaire mais elle semble bien meilleure que les trois autres méthodes sur ce jeu de données. La figure ci-dessous résume sur l'ensemble de l'échantillon test, la distribution des tailles d'intervalles de prédiction contenant vraimment ou non la vrai valeur selon les méthodes.

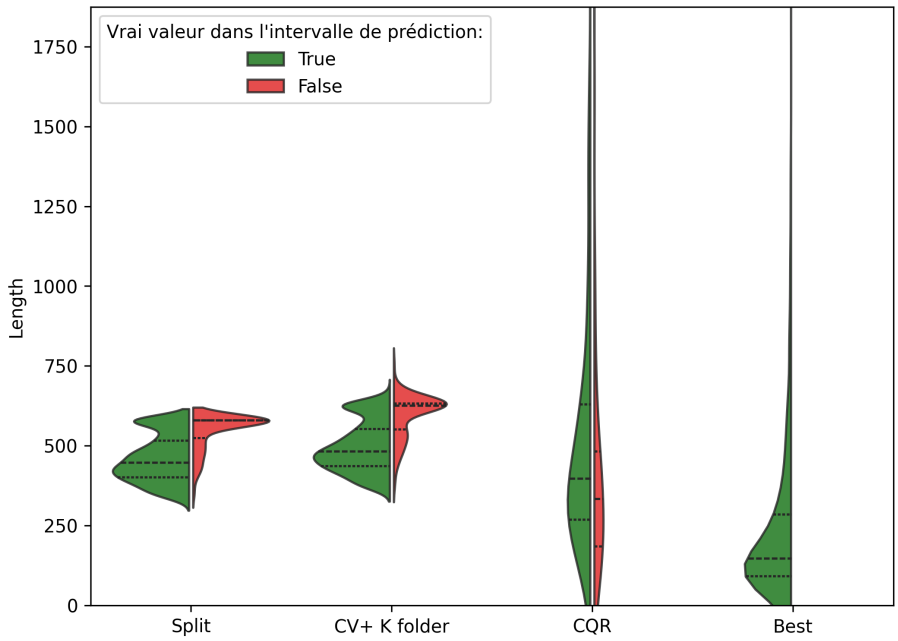


FIGURE 10 – Comparaison de la taille des intervalles

Conclusion

Dans le cadre de ce projet, nous avons exploré les possibilités offertes par les méthodes avancées de prédiction conforme et d'analyse de données textuelles, en utilisant comme vecteur principal les données issues de la base MIMIC-III. D'un point de vue analytique, nous avons mis en œuvre des techniques de pointe pour le traitement du langage naturel (NLP), notamment BERT, afin de convertir les données textuelles complexes des dossiers médicaux en vecteurs de caractéristiques exploitables. Sur le plan méthodologique, notre projet a intégré l'usage de la prédiction conforme pour fournir des estimations accompagnées d'intervalles de prédiction, offrant ainsi une meilleure appréciation des incertitudes associées aux prédictions. Cette approche, en enrichissant la prédiction classique par des modèles de régression ou de forêt aléatoire, propose un cadre plus robuste pour la prise de décision médicale, basée sur des données probantes et quantifiables. Néanmoins, notre travail n'est pas exempt de limites. La principale contrainte réside dans la complexité et la variabilité inhérente aux données médicales, notamment les notes cliniques qui requièrent des techniques d'analyse sophistiquées pour une exploitation optimale. Nous nous sommes également retrouvé confronté à des temps de calcul très long que nous avons réalisé sur la platforme Onyxia - SSP Cloud avec des sessions GPU . Ces temps de calculs ont été de l'ordre de 2 semaines pour la Figure 12 par exemple, ce qui nous à grandement ralenti dans notre exploration des multiples méthodes.

De plus, notre projet est une première exploration qui est voué à être adapté et amélioré pour correspondre au données du centre Léon Bérard. Ce qui posera des problèmes d'adaptation aux nouvelles données, d'autant plus que nous travaillons ici avec le modèle `Clinical BERT` qui est précisément entraîné sur les données de MIMIC-III.

Références

[1] Anastasios N. Angelopoulos and Stephen Bates (2022) *A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification*. Disponible sur : `https://arxiv.org/abs/2107.07511`

[2] I. Chevyrev and A. Kormilitzin, *A Primer on the Signature Method in Machine Learning*. In : arXiv :1603.03788 (2016).

[3] Y. Romano, E. Patterson, and E. Candès, *Conformalized quantile regression* in Advances in Neural Information Processing Systems, vol. 32, 2019, pp. 3543–3553

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert : Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv :1810.04805, 2018.

[5] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, *Predictive inference with the jack-knife+*, The Annals of Statistics, vol. 49, no. 1, pp. 486–507, 2021.

[6] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, Matthew B. A. McDermott (2019). *Publicly Available Clinical BERT Embeddings*. arXiv preprint arXiv :1904.03323.

[7] Nicolai Meinshausen (2006). *Quantile regression forests*. Journal of Machine Learning Research.

[8] Documentation MIMIC-III : *https ://mimic.mit.edu/docs/iii/tables/*

[9] Notre code : github.com/Krrcharles/Prediction-conforme-pour-les-donnees–analysee-par-Transformers

Annexes

Exemple de rapport médical

```
    54 y o man with DM c/b gastroparesis, neuropathy, ESRD s/p failed
transplant now on PD, severe 3V CAD, CHF, admitted yesterday after
presenting with weakness, confusion and hypotension following a run of
PD.
Wound infection
Assessment:
Pt with left 3rd toe dark/dusky, with reddened open area and red
streaking, also to heel of r foot likely cellulitis. Bilateral DP/PT
pulses palpable.
Action:
Vascular surgery took pt to OR for amputation of L toe. Pt on
Zosyn/Cipro for coverage.
Response:
Ongoing.
Plan:
Cont with abx. Hope is that pt
s wll improve once toe is amputated.
Renal failure, End stage (End stage renal disease, ESRD)
Assessment:
Pt with ESRD s/p renal transplant [**2119**] but requiring PD starting [**5-/2134**]
    **3-22**
failure of transplant.
Action:
Pt receiving Q4hr PD runs with 1500mLs 1.5% dextrose solution.
Response:
Pt tol PD runs.
Plan:
Cont PD as ordered per renal.
Sepsis, Severe (with organ dysfunction)
Assessment:
    **Name (NI) 11137**
pt on low dose levophed-0.03 mcg/kg/min. Venous o2 sat 73 and
lactate 1.8.
Action:
Levophed back on at 0.04 maps>70 to maintain venous sat closer to
normal range.
Response:
Lactate down ro 2, o2sat venous 67..
Plan:
Cont pt on low dose levo for now and trend mixed venous O2 sats and
lactate. If mixed venous O2 sat worsens, my consider starting
dobutamine. If improves, may wean levo OFF.
Hyperglycemia/Hypoglycemia
Assessment:
Pt with DMI, BSs in 50
s last eve. Improving to 150
Action:
Treated hypoglycemia with one amp d50.
Response:
Conts now to trend back down throughout the night
Plan:
Monitor BS Q6 and treat per SS. ???Cont fixed dose insulin in AM. If
hypoglycemic, treat with PRN dose dextrose.
```

Log-Signatures

La transformation de logsignature est une adaptation de la transformation de signature et peut être considérée comme une représentation réduite ou condensée de cette dernière. Alors que la signature d'un chemin génère un grand nombre de caractéristiques, potentiellement conduisant à une explosion dimensionnelle, la logsignature résume ces caractéristiques de manière plus compacte. Cette compression intelligente préserve les informations essentielles tout en réduisant la redondance, facilitant ainsi l'analyse et le traitement des séquences de données complexes sans compromettre la richesse des informations extraites.

L'intégration de la transformation de Logsignature dans les architectures d'apprentissage automatique offre une nouvelle dimension pour la modélisation et l'analyse des données séquentielles. En utilisant la logsignature comme prétraitement ou au sein même des réseaux neuronaux, on peut améliorer significativement la performance de divers modèles de machine learning, exploitant la capacité des transformations à encapsuler des informations complexes de manière efficace.

Log-Signatures vs Signatures

			Log-Signatures	Signatures
LR	Binaire	Taux de bien classés	48.12%	49.32%
		Rappel	46.00%	43.00%
		F1-score	15.09%	14.54%
RF	Binaire	Taux de bien classés	34.29%	38.90%
		Rappel	81.00%	79.00%
		F1-score	19.82%	20.59%
	Régression	MSE	56650	57373
		R^2	0.29	0.28

TABLE 6 – Comparaison entre les Log-Signatures ou les Signatures

Propriétés des signatures

Les signatures et les log-signatures présentent des propriétés intéressantes. Voici celles que nous utilisons dans notre projet :

1. L'invariance par rapport au temps :

   Soient $X, Y : [a,b] \to \mathbb{R}$ deux chemins à valeurs réelles et $\psi : [a,b] \to [a,b]$ une reparamétrisation. Définissons les chemins $\tilde{X}, \tilde{Y} : [a,b] \to \mathbb{R}$ par $\tilde{X}_t = X_{\psi(t)}$ et $\tilde{Y}_t = Y_{\psi(t)}$. Nous observons que

   $$\dot{\tilde{X}}_t = \dot{X}_{\psi(t)}\dot{\psi}(t),$$

   d'où

   $$\int_a^b \tilde{Y}_t d\tilde{X}_t = \int_a^b Y_{\psi(t)}\dot{X}_{\psi(t)}\dot{\psi}(t)dt = \int_a^b Y_u dX_u,$$

   En effectuant la substitution : $u = \psi(t)$. Cela montre que les signatures sont invariantes sous une reparamétrisation temporelle des deux chemins.

   Cette propriété nous est utile afin de pouvoir avoir des écarts de temps non régulier entre les rapports.

2. L'égalité de Chen :

   Soit $X : [a,b] \to \mathbb{R}^d$ et $Y : [b,c] \to \mathbb{R}^d$ deux chemins. Alors

   $$S(X * Y)_{a,c} = S(X)_{a,b} \otimes S(Y)_{b,c}.$$

   avec pour $\otimes$:

   $$e_{i_1} \dots e_{i_k} \otimes e_{j_1} \dots e_{j_m} = e_{i_1} \dots e_{i_k} e_{j_1} \dots e_{j_m}.$$

   Et pour $*$:

   Soit les chemins $X : [a,b] \to \mathbb{R}^d$ et $Y : [b,c] \to \mathbb{R}^d$, alors $X * Y : [a,c] \to \mathbb{R}^d$ a pour expression $(X * Y)_t = X_t$ pour $t \in [a,b]$ et $(X * Y)_t = X_b + (Y_t - Y_b)$ pour $t \in [b,c]$.

   Cette égalité nous permet lors du traitement des données de découper les rapports en parties avant de les donners au BERT. Cela nous permet de reduire grandement le temps de calcul des embeddings du BERT.

Full conformal

Cette méthode ne nécessite pas d'échantillon de calibration contrairement à la méthode **Split**. Comme $Y \in \mathscr{Y} \subseteq \mathbb{R}$, pour chaque $y \in \mathscr{Y}$, nous ajustons un nouveau modèle $\hat{f}_y$ sur l'ensemble de données augmenté $(X_1, Y_1), \ldots, (X_{n+1}, y)$. Il est important que l'ajustement du modèle pour $\hat{f}$ soit invariant aux permutations des données. Ensuite, nous calculons une fonction de score dont le modèle $\hat{f}_y$ est maintenant donné comme un argument car il n'est plus fixe : $s_{i,y} = s(X_i, Y_i, \hat{f}_y)$ pour $i = 1, \ldots, n$ et $s_{n+1,y} = s(X_{n+1}, y, \hat{f}_y)$. Ensuite, nous calculons le quantile conforme,

$$\hat{q}_y = \text{quantile}\left(s_{1,y}, \ldots, s_{n,y}; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right)$$

Enfin, nous rassemblons toutes les valeurs de y qui sont suffisamment cohérentes avec les données précédentes $(X_1, Y_1), \ldots, (X_n, Y_n)$ et les collectons dans un ensemble de confiance pour la valeur inconnue de Y_{n+1} :

$$C(X_{n+1}) = \{y : s_{n+1,y} \leq \hat{q}_y\}$$

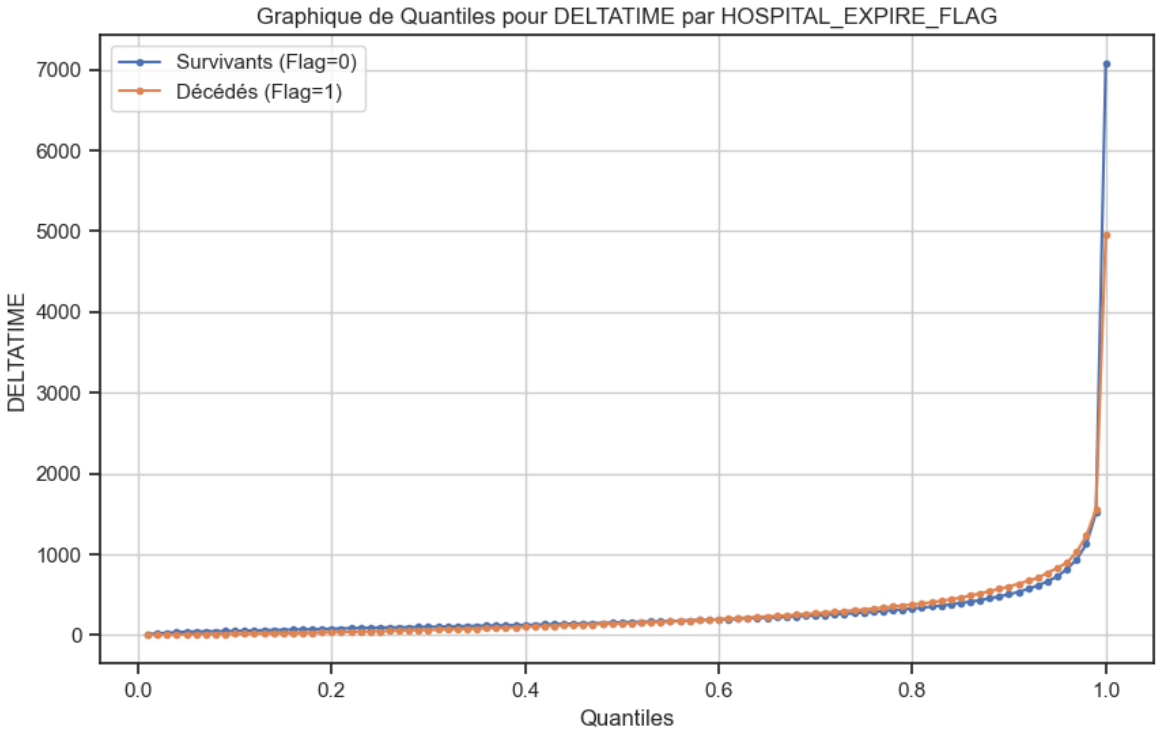


FIGURE 11 – Graphique de quantiles pour la durée de séjour

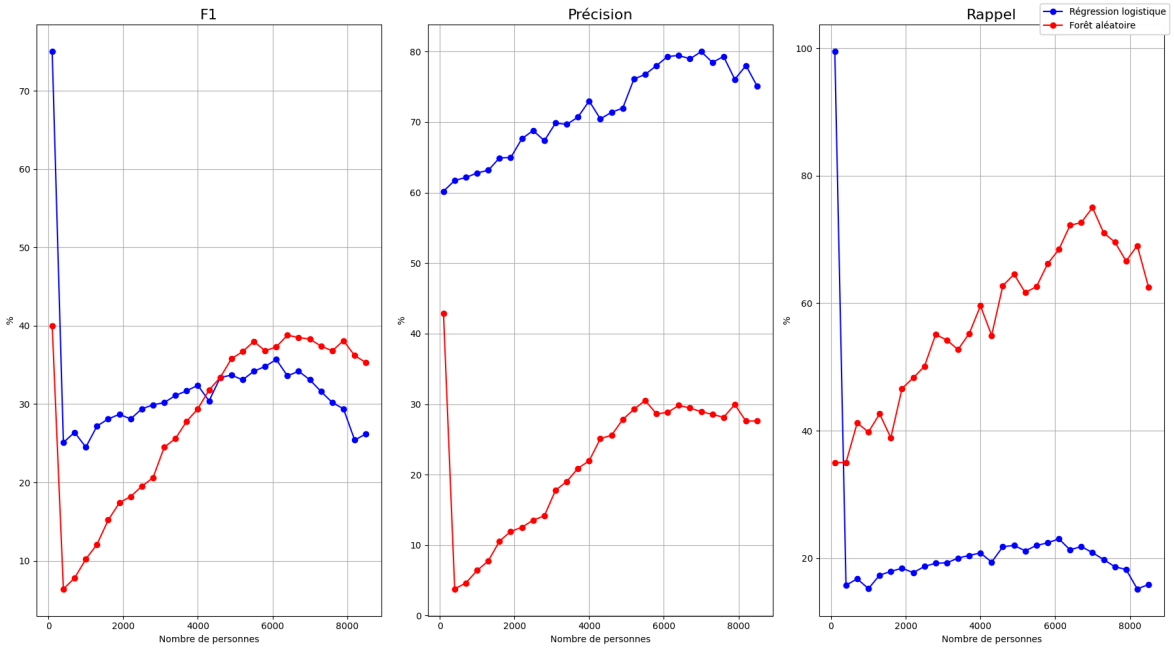
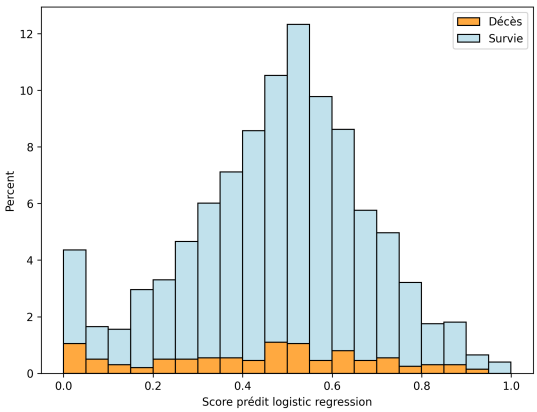
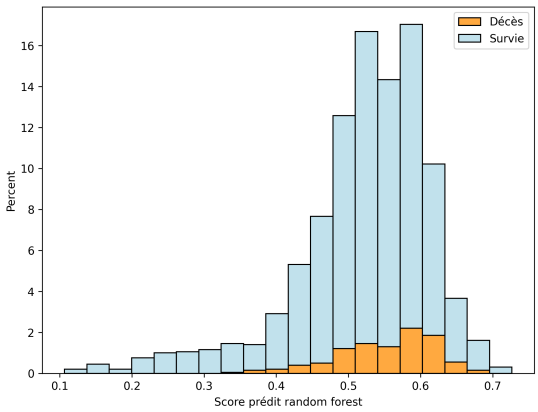


FIGURE 12 – Comparaison F1-score, précision et rappel selon le nombre d'observations

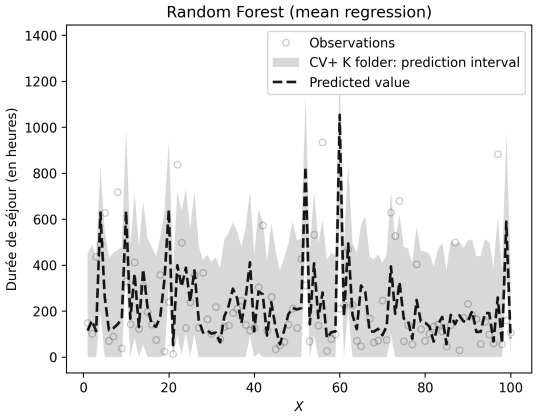


(a) Cas de la régression logistique

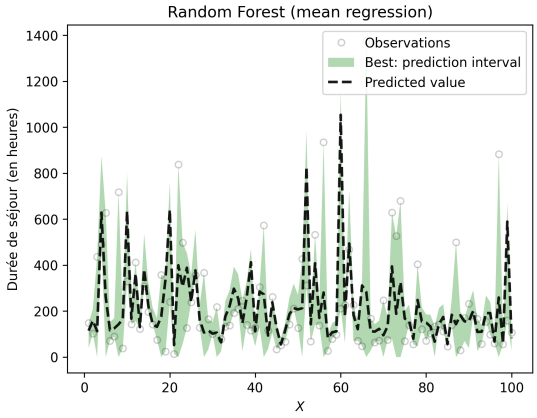


(b) Cas de la forêt aléatoire

FIGURE 13 – Histogramme des scores

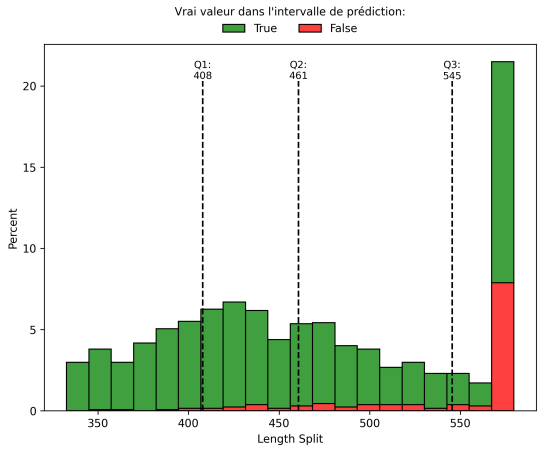


(a) Méthode CV+ à 5 plis

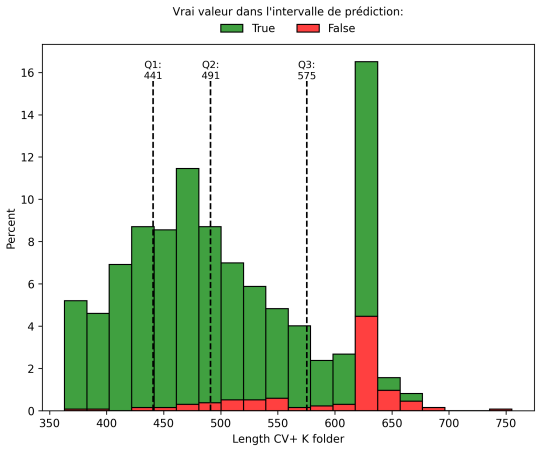


(b) Méthode 'Best' adaptée aux données

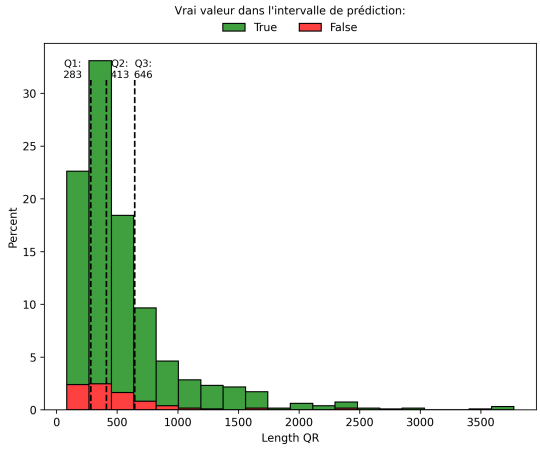
FIGURE 14 – Visualisation des intervalles de prédiction pour cent observations



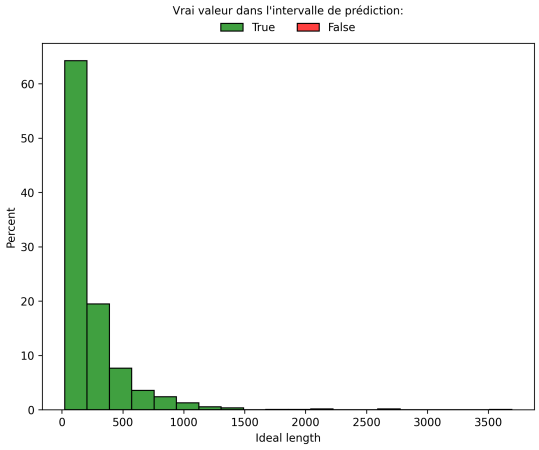
(a) Méthode Split



(b) Méthode CV+ à 5 plis



(c) Méthode CQR



(d) Méthode 'Best' adaptée aux données

FIGURE 15 – Histogrammes de la taille des intervalles de prédiction

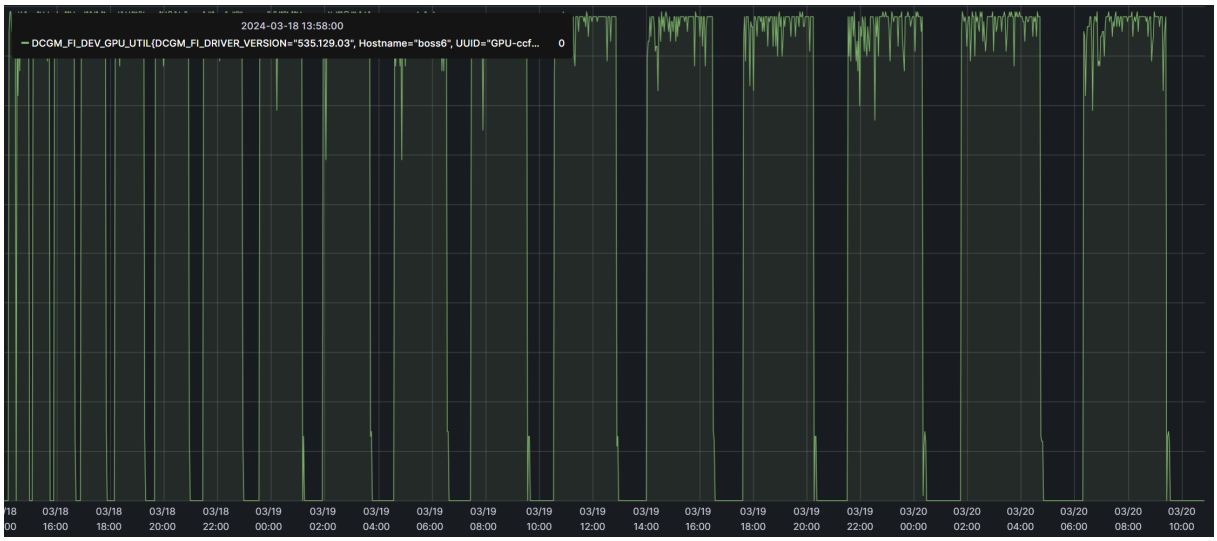


FIGURE 16 – Utilisation GPU durant une session

On remarque bien que la durée d'entrainement augmente fortement lorsque l'on met davantage de données d'entrainement.