Challenge QRT 2025

Jules Mourgues-Haroche, Alexandre Remiat

Mars 2025

Prédiction de Survie Globale de patients atteints de Leucémie Myéloïde

Contents

1 Introduction au Challenge

1.1 Contexte et objectifs

Au cours des dernières années, le secteur médical a connu une transformation majeure grâce à l'adoption de méthodes d'analyse de données massives. Ces techniques ont permis de développer des modèles prédictifs en santé, révolutionnant ainsi la manière dont les soins sont dispensés, notamment en oncologie. L'utilisation de ces approches permet d'affiner les stratégies thérapeutiques en personnalisant les traitements et en optimisant l'allocation des ressources, améliorant ainsi la qualité et le timing des décisions cliniques.

Dans ce contexte, le Data Challenge de QRT, organisé en partenariat avec l'Institut Gustave Roussy, vise à développer des modèles prédictifs capables d'estimer le risque de décès chez les patients diagnostiqués avec un sous-type de leucémie myéloïde adulte. L'objectif principal est de mesurer la survie globale — définie comme la période allant du diagnostic initial jusqu'au décès ou au dernier suivi — afin d'identifier de manière proactive les patients à risque élevé et de contribuer à des prises de décisions thérapeutiques mieux adaptées.

1.2 Présentation du problème

Le challenge consiste à concevoir et calibrer des modèles de survie pour prédire de manière fiable le pronostic des patients atteints de leucémie myéloïde. La problématique repose sur l'estimation précise de la survie globale, une métrique essentielle pour adapter l'approche thérapeutique : une prédiction erronée pourrait conduire à des traitements inadaptés, exposant les patients à des risques inutiles ou retardant l'accès à des interventions vitales.

Les participants auront à travailler avec un ensemble de données réelles provenant de 24 centres cliniques, intégrant des informations cliniques et moléculaires hétérogènes. Ils devront relever plusieurs défis, notamment la gestion des valeurs manquantes, l'extraction de features pertinentes et la calibration des modèles de survie. La réussite de ce challenge repose sur la capacité à transformer des données complexes en insights cliniquement exploitables, permettant ainsi d'améliorer significativement la prise en charge des patients.

2 Description des données

2.1 Présentation du Dataset

Le jeu de données est fourni sous la forme de deux fichiers ZIP et d'un fichier CSV. Plus précisément, les fichiers `X_train.zip` et `X_test.zip` contiennent respectivement les données d'entrée pour l'entraînement et le test, tandis que le fichier `Y_train.csv` regroupe les informations relatives aux temps de survie et aux statuts des patients. Le jeu d'entraînement comporte des données sur 3 323 patients, et le jeu de test sur 1 193 patients. Les données d'entrée sont organisées en deux ensembles complémentaires :

- **Données Cliniques** : informations médicales détaillées pour chaque patient.
- **Données Moléculaires** : informations génétiques concernant les mutations somatiques, chaque ligne correspondant à une mutation enregistrée pour un patient.

La colonne `ID` sert d'identifiant unique, permettant de relier les données cliniques, moléculaires et les informations de survie présentes dans `Y_train.csv`.

2.2 Objectif du Challenge et Format de Soumission

L'objectif du Data Challenge est de prédire la survie globale (OS) des patients diagnostiqués avec un sous-type de leucémie myéloïde adulte. Deux résultats clés sont fournis pour chaque patient dans le fichier `Y_train.csv` :

- **OS_YEARS** : Temps de survie global en années, calculé depuis le diagnostic.
- **OS_STATUS** : Indicateur d'état, où la valeur 1 indique un décès et 0 indique que le patient était vivant lors du dernier suivi.

La soumission doit être constituée d'un fichier CSV indexé par la colonne `ID` et comportant une colonne `risk_score` qui contient la prédiction du risque de décès pour chaque patient. Il est important de noter que, pour l'évaluation, seule l'ordre des risques prédits est pris en compte. Ainsi, si pour deux patients i et j, le risque prédit pour i est inférieur à celui de j, cela indique que l'on estime que i survivra plus longtemps que j.

2.3 Métrique d'évaluation : IPCW-C-index

L'évaluation des modèles se base sur l'IPCW-C-index, une extension du C-index traditionnel qui permet de gérer la censure à droite des données. Le C-index mesure la capacité d'un modèle à ordonner correctement les temps de survie, en calculant la proportion de paires comparables d'individus pour lesquelles les prédictions des risques de décès sont cohérentes avec les temps de survie réels. Dans le contexte de la censure à droite, certaines observations ne disposent pas du temps de survie complet (par exemple, lorsque le patient est toujours en vie lors du dernier suivi). L'IPCW-C-index ajuste alors le calcul en attribuant des poids inverses, basés sur la probabilité de censure, afin de compenser cette limitation. Cette métrique est implémentée dans la librairie `scikit-survival` et est tronquée à 7 ans pour le benchmark du challenge.

2.4 Contenu des Données Cliniques et Moléculaires

Les **Données Cliniques** se présentent sous forme d'une ligne par patient et incluent les informations suivantes :

- **ID** : Identifiant unique du patient.
- **CENTER** : Centre clinique de traitement.
- **BM_BLAST** : Pourcentage de blastes dans la moelle osseuse.
- **WBC** : Nombre de globules blancs (Giga/L).
- **ANC** : Nombre absolu de neutrophiles (Giga/L).
- **MONOCYTES** : Nombre de monocytes (Giga/L).
- **HB** : Taux d'hémoglobine (g/dL).
- **PLT** : Nombre de plaquettes (Giga/L).
- **CYTOGENETICS** : Description du caryotype, avec des notations conformes à la norme ISCN (par exemple, 46,XX ou 46,XY), et la détection d'anomalies telles que la monosomie 7.

Les **Données Moléculaires** fournissent, pour chaque mutation somatique identifiée chez un patient, les informations suivantes :

- **ID** : Identifiant unique du patient.

- **CHR, START, END** : Position chromosomique de la mutation.
- **REF, ALT** : Nucléotides de référence et alternatifs.
- **GENE** : Gène affecté par la mutation.
- **PROTEIN_CHANGE** : Impact de la mutation sur la protéine.
- **EFFECT** : Classification de l'impact fonctionnel de la mutation.
- **VAF** : Fraction allélique variante indiquant la proportion de cellules portant la mutation.

2.5 Benchmark fourni

Deux modèles de référence sont proposés dans le benchmark :

1. Un modèle simple LightGBM utilisant uniquement les données cliniques, sans gestion explicite de la censure.
2. Un modèle de risques proportionnels de Cox, intégrant à la fois les données cliniques et certaines informations sur les mutations génétiques.

Le score retenu pour le benchmark est celui obtenu avec le modèle de Cox, servant de référence pour évaluer les performances des modèles développés par les participants.

3 Analyse Exploratoire des Données (EDA)

Dans cette section, nous présentons de brefs résultats issus de l'analyse exploratoire des données cliniques et moléculaires. En plus de ces résultats, nous avons développé un rapport EDA à l'aide du package `AutoViz`, qui offre une visualisation plus complète de nos variables en fonction de nos variables cibles, à savoir `OS_STATUS` et `OS_YEARS`. Ce rapport peut être généré après la création de nos features et nous permet de mieux comprendre les interactions avec les variables cibles.

3.1 Données Cliniques

Le jeu de données cliniques comporte 3 323 patients et comprend plusieurs variables quantitatives et qualitatives telles que le pourcentage de blastes (`BM_BLAST`), le nombre de globules

blancs (WBC), l'hémoglobine (HB), le nombre de plaquettes (PLT) ainsi que des informations catégorielles comme le centre de traitement (CENTER) et le caryotype (CYTOGENETICS).

Statistiques descriptives. Le tableau 1 résume les principales statistiques descriptives pour les variables numériques.

Table 1: Statistiques descriptives des variables cliniques

Variable	Mean	Std	Skew	Min	25%	50%	75%	Max
BM_BLAST	5.98	7.62	3.62	0.00	1.00	3.00	8.00	91.00
WBC	6.54	10.25	7.09	0.20	2.70	4.10	6.66	154.40
ANC	3.26	5.24	8.07	0.00	1.00	2.00	3.69	109.62
MONOCYTES	0.96	2.67	8.94	0.00	0.15	0.37	0.78	44.20
HB	9.89	2.04	0.25	4.00	8.50	9.70	11.20	16.60
PLT	167.05	149.48	2.32	2.00	65.50	123.00	229.50	1451.00

Table 2: Valeurs manquantes par variable

Variables	Valeurs manquantes	Pourcentage
MONOCYTES	601	18.09%
CYTOGENETICS	387	11.65%
WBC	272	8.19%
ANC	193	5.81%
PLT	124	3.73%
HB	110	3.31%
BM_BLAST	109	3.28%

La heatmap de corrélation (Figure 1) met en évidence plusieurs relations notables: WBC, ANC et MONOCYTES présentent des coefficients de corrélation supérieurs à 0.75, reflétant leur interdépendance physiologique (toutes trois étant liées aux cellules sanguines). À l'inverse, BM_BLAST et PLT montrent un niveau de corrélation négatif modéré (-0.25), suggérant des mécanismes biologiques distincts.

La Figure 2 illustre à la fois les nuages de points (relationships pairwise) et les distributions (sur la diagonale) des principales variables cliniques. Nous constatons que BM_BLAST, WBC, ANC et MONOCYTES présentent une asymétrie marquée (skewness), principalement due à la présence de valeurs extrêmes. Pour mieux visualiser ces distributions, nous avons donc opté pour une représentation en échelle logarithmique sur le graphique de dispersion, ce qui permet de limiter l'impact des outliers et de mieux mettre en évidence les tendances globales. À l'inverse, des variables comme HB et PLT apparaissent moins asymétriques, bien que PLT conserve tout de même une certaine dispersion.

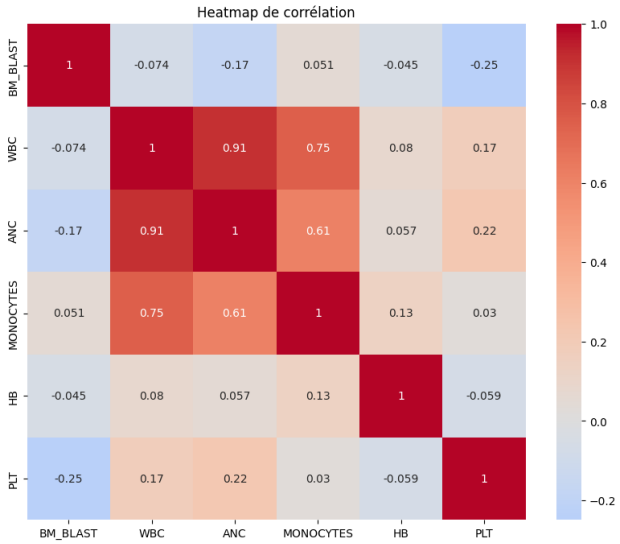

Figure 1: Heatmap de corrélation des données cliniques.

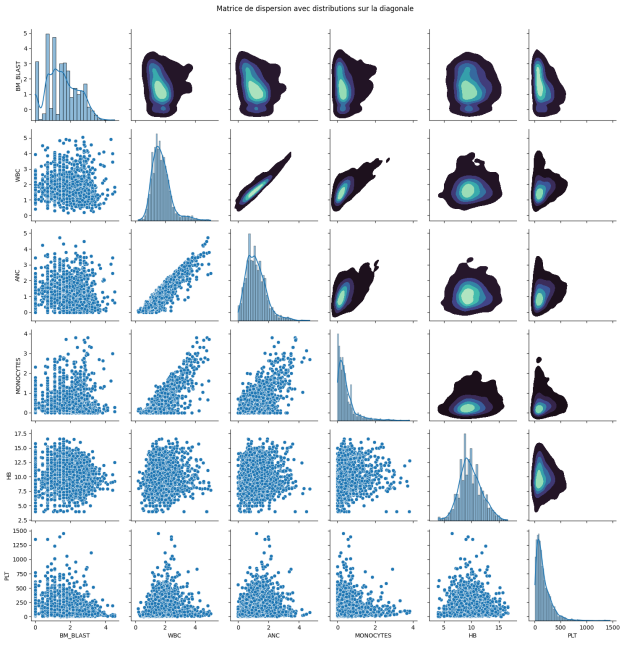

Figure 2: Matrice de dispertion des données cliniques avec distributions sur la diagonale.

Table 3: Centres et valeurs manquantes cliniques associées - Données d'entrainement

		Valeurs manquantes par CENTER						
CENTER	Proportion	BM_BLAST	WBC	ANC	MONOCYTES	HB	PLT	CYTOGENETICS
KI	27.08%	6.11%	5.67%	6.67%	9.00%	5.56%	5.67%	11.56%
DUS	13.69%	0.44%	30.77%	6.37%	40.22%	-	0.22%	30.77%
PV	9.51%	-	-	-	4.75%	-	-	6.65%
GESMD	7.40%	0.41%	-	0.41%	-	-	0.41%	4.88%
RMCN	5.99%	11.56%	9.05%	10.05%	45.23%	9.05%	9.05%	13.07%
CCH	4.78%	-	0.63%	0.63%	33.96%	0.63%	-	0.63%
CGM	3.22%	-	-	0.93%	-	-	-	-
ROM	3.13%	1.92%	2.88%	1.92%	1.92%	0.96%	0.96%	4.81%
UOB	2.65%	1.14%	1.14%	1.14%	5.68%	2.27%	1.14%	2.27%
HMS	2.50%	19.28%	40.96%	55.42%	55.42%	21.69%	40.96%	26.51%
MUV	2.50%	-	-	-	-	-	-	6.02%
TUD	2.20%	4.11%	2.74%	8.22%	8.22%	2.74%	2.74%	-
FUCE	2.20%	-	1.37%	1.37%	-	-	-	27.40%
ICO	2.14%	-	-	-	1.41%	-	-	2.82%
FLO	2.05%	-	-	-	5.88%	-	-	-
DUTH	1.99%	6.06%	22.73%	24.24%	25.76%	22.73%	18.18%	21.21%
UOXF	1.50%	2.00%	4.00%	8.00%	100.00%	4.00%	4.00%	2.00%
HIAE	1.41%	-	-	-	-	-	-	6.38%
MSK	1.11%	-	-	-	-	-	-	5.41%
IHBT	0.99%	-	-	6.06%	75.76%	-	-	15.15%
VU	0.99%	-	3.03%	-	-	-	-	-
UMG	0.78%	3.85%	11.54%	11.54%	61.54%	3.85%	3.85%	3.85%
REL	0.18%	-	-	-	100.00%	-	-	16.67%

Table 4: Centres et valeurs manquantes cliniques associées - Données d'évaluation

		Valeurs manquantes par CENTER						
CENTER	Proportion	BM_BLAST	WBC	ANC	MONOCYTES	HB	PLT	CYTOGENETICS
KYW	100.0%	9.64%	9.39%	11.82%	74.02%	9.30%	9.64%	9.72%

Les Tableaux 3 et 4 mettent en évidence d'importantes disparités dans la répartition des valeurs manquantes selon les centres. Par exemple, certains centres (comme `HMS` ou `UOXF` dans les données d'entraînement) présentent un taux de données manquantes nettement plus élevé pour plusieurs variables, notamment `MONOCYTES`, `WBC` ou `ANC`. Ces écarts peuvent s'expliquer par des pratiques de mesure ou de saisie différentes d'un centre à l'autre, ou encore par des contextes cliniques spécifiques. Cette hétérogénéité doit être prise en compte lors de la phase de nettoyage et d'imputation des données, afin de ne pas biaiser les analyses ultérieures.

3.2 Données Moléculaires

Le jeu de données moléculaires contient les informations relatives aux mutations somatiques observées chez les patients. Chaque ligne correspond à une mutation identifiée dans les

cellules cancéreuses et fournit des détails essentiels sur la position génomique ainsi que sur l'impact fonctionnel de la mutation. Les principales variables enregistrées incluent les colonnes `CHR`, `START` et `END` qui indiquent la position chromosomique de la mutation dans le génome humain, `REF` et `ALT` qui précisent respectivement le nucléotide de référence et le nucléotide alternatif (mutant), et `GENE` qui désigne le gène affecté par la mutation. De plus, la variable `PROTEIN_CHANGE` décrit l'impact de la mutation sur la protéine produite, tandis que `EFFECT` classe l'effet fonctionnel de la mutation sur le gène. Les colonnes `VAF` et `DEPTH` fournissent, respectivement, la fraction allélique variante, qui représente la proportion de cellules présentant la mutation, et la profondeur de séquençage au niveau de la mutation. Ces données moléculaires permettent d'explorer l'hétérogénéité génétique des tumeurs et d'approfondir la compréhension du profil mutationnel des patients en complément des informations cliniques.

Table 5: Statistiques descriptives des variables moléculaires

Variable	Mean	Std	Skew	Min	25%	50%	75%	Max
START	8.0783×10^7	5.6427×10^7	0.5842	3.9490×10^5	3.1022×10^7	7.4733×10^7	1.1526×10^8	2.2625×10^8
END	8.0783×10^7	5.6427×10^7	0.5842	3.9490×10^5	3.1022×10^7	7.4733×10^7	1.1526×10^8	2.2625×10^8
VAF	0.3051	0.2115	0.6607	0.0200	0.1026	0.3213	0.4420	0.9990
DEPTH	1051.23	552.86	1.2530	16.00	660.00	975.00	1353.00	7156.00

Table 6: Valeurs manquantes par variable

Variables	Valeurs manquantes	Pourcentage
CHR	114	1.04%
START	114	1.04%
END	114	1.04%
REF	114	1.04%
ALT	114	1.04%
DEPTH	114	1.04%
VAF	89	0.81%
PROTEIN_CHANGE	12	0.11%

La variable `VAF` (Variant Allele Fraction) se situe entre 0 et 1, avec une légère asymétrie (skewness) mise en évidence dans les statistiques descriptives (Table 5). Quant à `DEPTH`, elle présente également une distribution étendue et un skewness notable, avec quelques valeurs élevées qui tirent la moyenne vers le haut. La heatmap de corrélation (Figure 3) montre une corrélation négative très faible entre `VAF` et `DEPTH`, suggérant qu'une plus grande profondeur de séquençage ne se traduit pas forcément par une fraction allélique plus ou moins élevée, mais peut apporter davantage de confiance dans la détection de la mutation.

Par ailleurs, la Figure 4 met en évidence la répartition et les distributions des principales variables moléculaires. On observe que `START` et `END` sont extrêmement corrélées (voir

également la Figure 3), ce qui s'explique par le fait qu'elles désignent respectivement les positions de début et de fin d'une même mutation sur le génome. Dans la majorité des cas, la longueur de l'intervalle (`END - START`) est relativement faible, ce qui se traduit par une forte corrélation entre ces deux variables. En pratique, il peut être plus pertinent de travailler directement avec la différence `END - START` comme indicateur de la taille de la région altérée, plutôt que de conserver simultanément `START` et `END` qui introduisent une forte redondance.

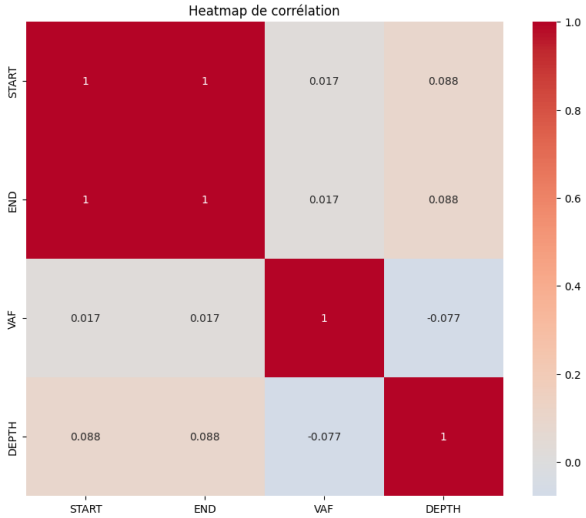

Figure 3: Heatmap de corrélation des données moléculaires.

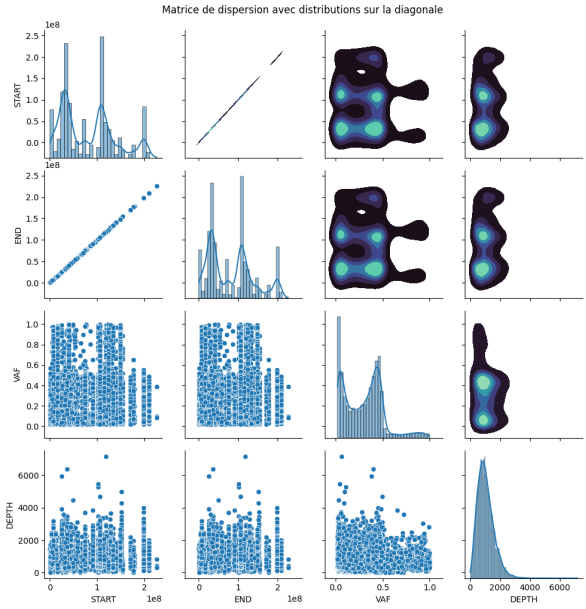

Figure 4: Matrice de dispertion des données moléculaires avec distributions sur la diagonale.

Table 7: Comparaison des pourcentages de valeurs manquantes par CENTER pour les données moléculaires d'entraînement

		Valeurs manquantes par CENTER									
CENTER	Proportion	CHR	START	END	REF	ALT	GENE	PROTEIN_CHANGE	EFFECT	VAF	DEPTH
KI	26.07%	1.05%	1.05%	1.05%	1.05%	1.05%	-	0.21%	-	0.88%	1.05%
DUS	14.13%	1.17%	1.17%	1.17%	1.17%	1.17%	-	0.06%	-	0.71%	1.17%
PV	9.76%	0.56%	0.56%	0.56%	0.56%	0.56%	-	0.19%	-	0.37%	0.56%
GESMD	8.06%	0.68%	0.68%	0.68%	0.68%	0.68%	-	-	-	0.45%	0.68%
RMCN	6.30%	0.58%	0.58%	0.58%	0.58%	0.58%	-	-	-	0.58%	0.58%
CCH	4.09%	3.13%	3.13%	3.13%	3.13%	3.13%	-	-	-	2.91%	3.13%
ROM	3.07%	0.60%	0.60%	0.60%	0.60%	0.60%	-	-	-	0.60%	0.60%
MUV	2.87%	1.91%	1.91%	1.91%	1.91%	1.91%	-	-	-	0.96%	1.91%
HMS	2.86%	2.56%	2.56%	2.56%	2.56%	2.56%	-	-	-	2.24%	2.56%
UOB	2.78%	0.99%	0.99%	0.99%	0.99%	0.99%	-	-	-	0.33%	0.99%
CGM	2.71%	0.68%	0.68%	0.68%	0.68%	0.68%	-	0.34%	-	0.68%	0.68%
DUTH	2.29%	0.40%	0.40%	0.40%	0.40%	0.40%	-	-	-	0.40%	0.40%
ICO	2.16%	0.42%	0.42%	0.42%	0.42%	0.42%	-	-	-	0.42%	0.42%
FUCE	2.09%	0.87%	0.87%	0.87%	0.87%	0.87%	-	-	-	0.87%	0.87%
FLO	1.96%	-	-	-	-	-	-	-	-	-	-
HIAE	1.78%	0.51%	0.51%	0.51%	0.51%	0.51%	-	-	-	0.51%	0.51%
TUD	1.75%	1.05%	1.05%	1.05%	1.05%	1.05%	-	0.52%	-	1.05%	1.05%
UOXF	1.27%	-	-	-	-	-	-	0.72%	-	-	-
MSK	1.09%	1.68%	1.68%	1.68%	1.68%	1.68%	-	-	-	1.68%	1.68%
IHBT	0.97%	4.72%	4.72%	4.72%	4.72%	4.72%	-	-	-	2.83%	4.72%
VU	0.97%	0.94%	0.94%	0.94%	0.94%	0.94%	-	-	-	0.94%	0.94%
UMG	0.82%	-	-	-	-	-	-	-	-	-	-
REL	0.16%	-	-	-	-	-	-	-	-	-	-

Table 8: Centres et valeurs manquantes moléculaires associées - Données d'évaluation

		Valeurs manquantes par CENTER									
CENTER	Proportion	CHR	START	END	REF	ALT	GENE	PROTEIN_CHANGE	EFFECT	VAF	DEPTH
KYW	100.0%	2.23%	2.23%	2.23%	2.23%	2.23%	0.00%	1.49%	2.91%	0.0%	2.23%

Comme l'indiquent les Tableaux 6 et 7, la proportion de valeurs manquantes (˜1 %) reste globalement faible pour les variables `CHR`, `START`, `END`, `REF`, `ALT`, `DEPTH` et `VAF`. Toutefois, certains centres affichent des taux de données manquantes plus élevés pour certaines variables (par exemple `IHBT`), ce qui peut s'expliquer par des différences dans les protocoles de séquençage ou de saisie. Une attention particulière devra être portée à ces disparités lors des étapes de nettoyage et d'imputation des données, afin de limiter l'introduction de biais dans les analyses ultérieures.

4 Modèle de Survie

4.1 Présentation du cadre théorique

Les modèles de survie visent à prédire le temps jusqu'à la survenue d'un événement d'intérêt (par exemple, le décès, la rechute ou toute autre issue cliniquement pertinente). Dans ce cadre, le problème est traité par des méthodes spécifiques qui tiennent compte des données censurées. Parmi les approches populaires, on trouve le modèle de Cox (Cox Proportional Hazards), une approche linéaire, mais également des méthodes d'ensembles, comme les forêts aléatoires ou des modèles de boosting qui permettent une approche non linéaire, et qui se révèlent particuliairement efficace si on les adaptent à l'analyse de survie.

4.2 Modèle Cox

Le modèle de Cox (Cox Proportional Hazards) est un modèle non-paramétrique permettant d'estimer l'effet des covariables sur le taux de risque, sans faire d'hypothèse sur la forme de la fonction de risque de base. Le modèle s'exprime à travers la fonction de risque :

$$h(t \mid X) = h_0(t) \exp\left(\beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p\right)$$

où :

- $h(t \mid X)$ est le taux de risque à l'instant t pour un individu possédant le vecteur de covariables X.
- $h_0(t)$ est le risque de base, qui reste non spécifié.
- $\beta_1, \beta_2, \ldots, \beta_p$ sont les coefficients associés aux covariables $X_1, X_2, \ldots, X_p$.

Le modèle estime les coefficients β de manière à maximiser la vraisemblance partielle, qui est définie tel que :

$$L(\beta) = \prod_{i:\delta_i=1} \frac{\exp\left(\beta^\top X_i\right)}{\sum_{j \in R(t_i)} \exp\left(\beta^\top X_j\right)}$$

où δ_i indique si l'événement a été observé pour l'individu i et $R(t_i)$ représente l'ensemble des individus à risque au moment t_i.

4.3 Random Survival Forest

La méthode des *Random Survival Forests* repose sur la construction d'un grand nombre d'arbres, chacun étant généré à partir d'un échantillon bootstrap des données d'apprentissage.

À chaque nœud de chaque arbre, une sélection aléatoire d'un sous-ensemble de variables et de seuils est effectuée afin d'optimiser la séparation des individus selon leur risque, souvent à l'aide d'une statistique de log-rank. Cette démarche, en plus de favoriser la dé-corrélation entre les arbres, améliore la robustesse du modèle, notamment dans le contexte d'une analyse de survie où les données censurées et la dimension temporelle des événements doivent être prises en compte.

Lorsqu'une observation est descendue dans chaque arbre jusqu'à atteindre un nœud terminal, les individus qui s'y retrouvent sont utilisés pour estimer de manière non paramétrique la fonction de survie ou le risque cumulatif. Deux estimateurs sont alors particulièrement pertinents : l'estimateur de Kaplan–Meier et l'estimateur de Nelson–Aalen.

4.3.1 Estimateur de Kaplan–Meier

L'estimateur de Kaplan–Meier permet d'estimer la fonction de survie $S(t)$, c'est-à-dire la probabilité qu'un individu ne subisse pas l'événement étudié (par exemple, un décès) au-delà d'un instant t. La formule de l'estimation est donnée par :

$$S(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right),$$

où :

- t_i représente les instants où un événement est observé,
- d_i est le nombre d'événements survenant à l'instant t_i,
- n_i désigne le nombre d'individus à risque immédiatement avant t_i.

Cette méthode construit une courbe en escalier : la fonction de survie reste constante entre deux instants d'événements et chute brusquement dès qu'un ou plusieurs événements surviennent. Les individus censurés sont inclus dans le calcul de n_i jusqu'à leur date de censure, ce qui permet de traiter correctement l'information incomplète.

4.3.2 Estimateur de Nelson–Aalen

L'estimateur de Nelson–Aalen se concentre quant à lui sur le risque cumulatif $H(t)$, qui reflète l'accumulation du risque jusqu'au temps t. La formule de cet estimateur est :

$$H(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}.$$

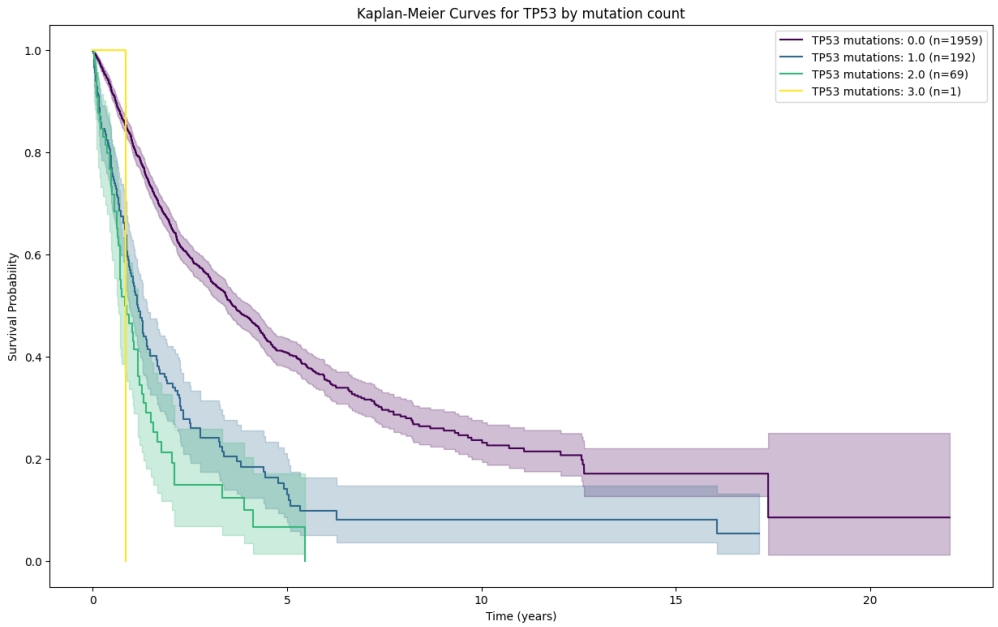

Figure 5: Fonction de survie selon Kaplan–Meier pour le gène TP53.

Le lien entre le risque cumulatif et la fonction de survie est établi par la relation suivante :

$$S(t) = \exp\left(-H(t)\right).$$

Ainsi, après avoir estimé $H(t)$ par la somme des risques instantanés $\frac{d_i}{n_i}$ sur les instants où survient un événement, on peut déduire la fonction de survie par une transformation exponentielle négative.

4.3.3 Agrégation des arbres

L'estimation finale de la fonction de survie dans une Random Survival Forest est obtenue par agrégation des prédictions issues de chacun des arbres. Si l'on note $\hat{S}_b(t \mid x)$ la fonction de survie estimée par l'arbre b et B le nombre total d'arbres, la fonction de survie globale est donnée par :

$$\hat{S}(t \mid x) = \frac{1}{B} \sum_{b=1}^{B} \hat{S}_b(t \mid x).$$

Cette agrégation permet de lisser les estimations locales issues de chaque arbre, renforçant ainsi la robustesse et la précision de la prédiction globale de la survie.

4.3.4 Complémentarité des approches

Les estimateurs de Kaplan–Meier et de Nelson–Aalen reposent sur des méthodes non paramétriques et tiennent compte des données censurées, mais ils offrent des points de vue complémentaires

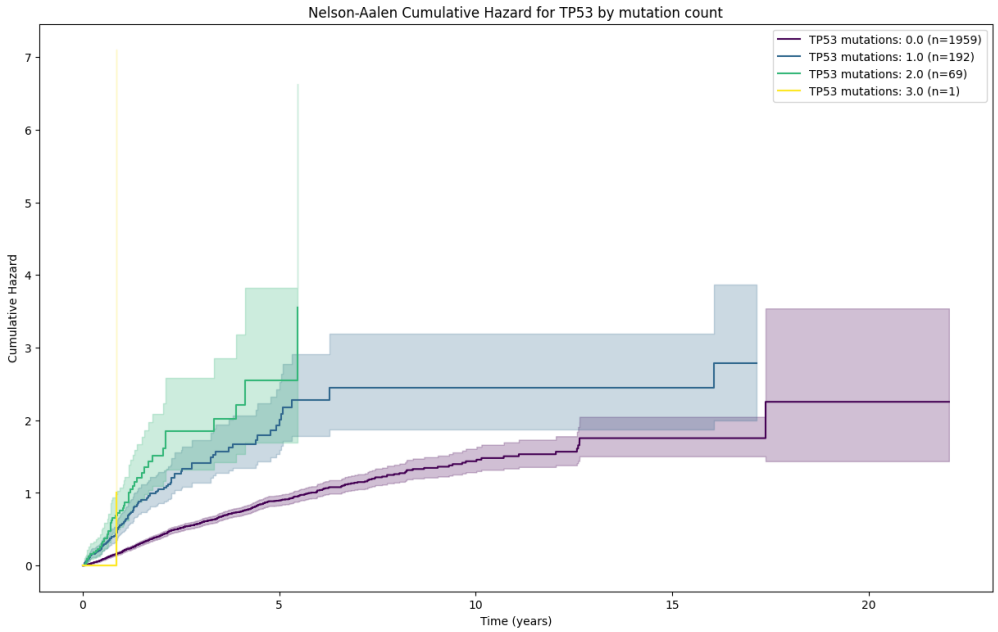

Figure 6: Fonction de survie selon Nelson–Aalen pour le gène TP53.

:

- **Kaplan–Meier** se focalise sur l'estimation directe de la probabilité de survie. Sa courbe en escalier fournit une représentation intuitive de la survie des individus dans le temps.
- **Nelson–Aalen** met en avant le risque cumulatif, permettant d'identifier l'accumulation du risque et son intensification au fil du temps, la transformation exponentielle fournissant ensuite une estimation de la survie.

Dans le cadre des Random Survival Forests, chaque nœud terminal peut ainsi recourir à l'un ou l'autre de ces estimateurs pour produire une prédiction robuste de la survie ou du risque, sans nécessiter de supposer une forme fonctionnelle préétablie pour le risque de base.

4.4 Gradient Boosting Survival Analysis

La méthode de Gradient Boosting Survival Analysis repose sur la combinaison itérative de modèles faibles pour optimiser une fonction de coût adaptée aux données de survie censurées. Chaque nouvel arbre se concentre sur les erreurs résiduelles des arbres précédents dans une approche séquentielle et gloutonne (*stagewise*). Ce cadre polyvalent ne correspond pas à un modèle particulier, mais permet d'optimiser de nombreuses fonctions de coût en combinant les prédictions de multiples apprenants de base, souvent très simples (qualifiés de *weak learners*) et à peine meilleurs qu'un tirage aléatoire.

Le modèle final est construit de manière additive selon la formule :

$$f_M(x) = f_0(x) + \sum_{m=1}^{M} \gamma_m \, h_m(x), \tag{1}$$

où

- $f_0(x)$ est le modèle initial (souvent une constante),
- $h_m(x)$ représente le modèle faible (par exemple, un arbre de décision) appris à l'itération m et paramétré par un vecteur θ_m,
- γ_m est le taux d'apprentissage qui pondère l'apport de chaque nouvel apprenant,
- M est le nombre total d'itérations (ou d'apprenants de base).

Un modèle de Gradient Boosting Survival est similaire à une Random Survival Forest dans la mesure où il s'appuie sur plusieurs apprenants de base pour produire une prédiction globale, mais il diffère dans la manière dont ces apprenants sont combinés. En effet, alors qu'une Random Survival Forest construit indépendamment un ensemble d'arbres de survie et en fait ensuite la moyenne des prédictions, le modèle en Gradient Boosting est construit de manière séquentielle.

La fonction de coût peut être spécifiée via l'argument `loss` ; ici, la fonction de coût par défaut est la log-vraisemblance partielle du modèle de Cox (coxph). L'objectif est de maximiser cette log-vraisemblance, ce qui revient en pratique à minimiser la fonction de perte définie comme l'opposé de la log-vraisemblance partielle :

$$L(f) = -\sum_{i:\delta_i=1} \left(f(x_i) - \log \sum_{j \in R(t_i)} \exp(f(x_j)) \right),$$

où δ_i est une variable indicatrice (1 si l'événement est observé, 0 en cas de censure) et $R(t_i)$ représente l'ensemble des individus à risque à l'instant t_i.

L'optimisation se fait de manière itérative en calculant, à chaque itération m, les résidus pseudo, c'est-à-dire les dérivées négatives de la fonction de coût par rapport aux prédictions courantes :

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=f_{m-1}(x)}. \tag{2}$$

Ces résidus, qui indiquent dans quelle mesure la prédiction actuelle $f_{m-1}(x)$ sous-estime ou surestime le risque, servent de cible pour l'apprentissage du modèle faible $h_m(x)$. Ainsi, chaque nouvel arbre est entraîné pour prédire ces résidus, et le modèle est mis à jour selon :

$$f_m(x) = f_{m-1}(x) + \gamma_m \, h_m(x). \tag{3}$$

Ce processus itératif permet de corriger progressivement les erreurs résiduelles et d'améliorer la capacité du modèle à prédire le risque de survie.

5 Traitement des Features

Dans notre approche, le traitement des features s'inscrit dans une démarche modulaire permettant d'exploiter et de fusionner nos deux sources principales de données : les données cliniques, contenant les informations patients, et les données moléculaire, recensant les mutations observées. Cette modularité repose sur trois listes de paramètres distincts :

- La liste `clinical` définit les opérations de prétraitement à appliquer aux données cliniques (exemple : traitement de la colonne cytogénétique, encodage de variables catégorielles, création de ratios ou d'interactions entre variables).
- La liste `molecular` spécifie les transformations à réaliser sur les données moléculaires (exemple : encodage de la variable `GENE`, extraction d'informations sur l'effet des mutations).
- La liste `merge` contrôle la méthode de fusion des données moléculaires sur les données cliniques. Selon la configuration, la fusion peut s'appuyer sur des outils comme `Featuretools` pour la synthèse automatique de nouvelles features.

La fonction principale de prétraitement orchestre l'ensemble de ces opérations. Elle commence par traiter indépendamment chaque source de données en fonction des paramètres spécifiés, puis procède à la fusion des informations selon la méthode choisie. Cette structure modulaire facilite l'intégration de nouvelles sources de données ou de nouvelles techniques de feature engineering sans modifier l'architecture globale du pipeline.

5.1 Exemples des différents traitements

5.1.1 Cytogénétique

Le pipeline intègre une transformation de la colonne `CYTOGENETICS` qui respecte les standards de notation ISCN. Par exemple, considérons la chaîne suivante :

```
45,xx,dic(5;17)(q13;p11.2),add(7)(q11.2),der(16)t(1;16)(p22;q24),inc[cp9]
/46,xx[11]
```

Cette description représente deux sous-clones cytogénétiques :

- **Premier sous-clone** : `45,xx,dic(5;17)(q13;p11.2),add(7)(q11.2), der(16)t(1;16)(p22;q24),inc[cp9]`.

- **Deuxième sous-clone** : `46,xx[11]`.

La fonction `parse_cytogenetics` décompose cette description et calcule plusieurs features utiles :

- **num_subclones** : nombre de sous-clones détectés. Ici, 2.
- **sex** : déterminé à partir de l'indication du sexe dans la chaîne (la présence de `xx` indique le sexe féminin, codé ici par 0).
- **avg_chromosomes** : moyenne du nombre de chromosomes des sous-clones. Avec 45 chromosomes pour le premier sous-clone et 46 pour le deuxième, on obtient $\frac{45+46}{2} = 45.5$.
- **total_mitoses** : somme des valeurs extraites entre crochets. Dans cet exemple, seul le deuxième sous-clone fournit une valeur numérique (`[11]`), donc le total est 11.
- **num_translocations** : le pipeline détecte une translocation dans le segment `der(16)t(1;16)(p22;q24)`, ce qui donne 1.
- **num_deletions**, **num_inversions** et **num_duplications** : aucune anomalie de ce type n'est détectée (0).
- **num_additions** : une addition est identifiée dans `add(7)(q11.2)`, soit 1.
- **num_monosomies** et **num_trisomies** : aucune anomalie du type monosomie ou trisomie n'est détectée (0).
- **complexity_score** : somme des anomalies structurelles (translocations, suppressions, inversions, duplications, additions). Ici, $1 + 0 + 0 + 0 + 1 = 2$.

Le tableau ci-dessous résume les résultats obtenus pour cette transformation :

5.1.2 Encodage

encodage des variables catégorielles à l'aide d'un encodage one-hot, avec la spécificité de pouvoir définir une variable `UNKNOWN` qui peut regrouper les catégories qui sont définies sous un certain seuil d'apparition.

Feature	Valeur
num_subclones	2
sex	0
avg_chromosomes	45.5
total_mitoses	11
num_translocations	1
num_deletions	0
num_inversions	0
num_duplications	0
num_additions	1
num_monosomies	0
num_trisomies	0
complexity_score	2

Table 9: Résultats de la transformation de la colonne `CYTOGENETICS` pour l'exemple donné.

5.1.3 Featuretools

L'utilisation de `Featuretools` permet de générer automatiquement de nouvelles features, la fonction va se charger de calculer les différentes features en fonction de leur type, voici les différentes features possibles : `sum`, `std`, `max`, `skew`, `min`, `mean`, `count`, `percent_true`, `num_unique`, `mode`.

5.1.4 Opérations

Le pipeline intègre des opérations de feature engineering sur le DataFrame fusionné, telles que la création de nouvelles variables par l'addition, la multiplication, le calcul de ratios ou de logarithmes, définies dynamiquement en fonction des opérateurs présents dans les paramètres cliniques et moléculaires.

5.1.5 Imputation et Winsorisation des Outliers

Le pipeline intègre également une étape de traitement des valeurs manquantes et des outliers. Pour les valeurs manquantes (NaN), une stratégie d'imputation peut être spécifiée, par exemple en remplaçant les NaN par la moyenne ou la médiane de la variable.

Par ailleurs, une procédure de winsorisation peut être appliquée pour limiter l'influence des outliers extrêmes. Pour chaque variable, le pipeline calcule les quantiles inférieurs et supérieurs à un seuil défini (par exemple, threshold = 0.01) de la manière suivante :

Soit $X = \{x_1, x_2, \ldots, x_n\}$ un ensemble de données. Pour winsoriser ces données, on

procède comme suit :

1. Définir le quantile inférieur q_1 et le quantile supérieur q_3 à partir d'un seuil τ (par exemple, $\tau = 0.01$) :

$$q_1 = \inf\{x \in \mathbb{R} : F_X(x) \geq \tau\}, \quad q_3 = \inf\{x \in \mathbb{R} : F_X(x) \geq 1 - \tau\},$$

où F_X est la fonction de répartition empirique de X.

2. Calculer l'intervalle interquartile (IQR) :

$$IQR = q_3 - q_1.$$

3. Définir les bornes inférieure et supérieure avec un multiplicateur k (souvent $k = 1.5$) :

$$L = q_1 - k \times IQR, \quad U = q_3 + k \times IQR.$$

4. Pour chaque observation $x \in X$, la valeur winsorisée x^* est donnée par :

$$x^* = \min\left(\max(x, L), U\right).$$

Ainsi, toute valeur inférieure à L est remplacée par L et toute valeur supérieure à U est remplacée par U.

5.1.6 Inférence du Sexe

Il peut arriver que la notation ISCN ne permette pas de déterminer de manière explicite le sexe d'un individu, notamment en l'absence d'information ou en présence d'ambiguïtés dans la chaîne de caractères. Pour pallier cette incertitude, le pipeline implémente une approche basée sur le bagging. Cette méthode nous permet d'utiliser l'ensemble des informations des autres variables pour fournir une estimation sur le sexe de l'individu lorsque l'information est manquante ou indéfinie. En pratique, lorsque la donnée relative au sexe est manquante, le pipeline attribue une valeur par agrégation pour garantir que chaque patient dispose d'une indication de sexe dans le jeu de données final.

5.2 Ajout de nouvelles features via l'API MyVariant

Pour compléter et enrichir le jeu de données, une étape d'extraction d'information supplémentaire est rendue possible via l'API `MyVariant`. Cette intégration permet de récupérer des annotations biologiques et des scores de prédiction (par exemple, CADD, Phred, rawscore, etc.) associés aux mutations répertoriées dans le données moléculaires.

Il est possible de récupérer ces données grâce aux colonnes `CHR`, `START`, `ALT` et `REF`, en combinant ces données au format `chr{CHR}:g.{START}{REF}<{ALT}` ce qui donne par exemple : `chrX:g.119388574C>G`. C'est avec cet identifiant qu'on est capable d'aller récupérer un ensemble d'informations supplémentaires sur la mutation.

Ce processus nous permet de rajouter une quatrième liste de paramètres nommée `additional`.

6 Sélection des Features

Dans le cadre de notre pipeline de modélisation, nous avons mis en place un rapport interactif au format HTML basé sur les méthodes SHAP (SHapley Additive exPlanations). Ce rapport permet d'examiner en détail l'impact de chaque feature sur les prédictions du modèle choisi. Il comprend plusieurs graphiques explicatifs, tels que :

- **Graphiques de résumé (summary plots)** : Ces visualisations montrent la distribution des valeurs SHAP pour chaque feature et permettent d'identifier rapidement celles qui ont le plus d'influence sur le modèle.
- **Top features et dépendance plots** : Nous sélectionnons les principales features (par exemple, le top 10) et générons des graphiques de dépendance entre elles. Ces visualisations permettent d'observer la linéarité ou non des relations entre les features, offrant ainsi une meilleure compréhension de leurs interactions.

De plus, lorsque le modèle fournit une mesure de l'importance des features (comme c'est le cas pour le Gradient Boosting Survival Analysis), ces informations sont également intégrées dans le rapport. Cela offre une double perspective : d'une part, l'analyse fine via SHAP qui décompose l'impact de chaque feature sur chaque prédiction, et d'autre part, une vue globale de l'importance des variables.

Cette approche nous permet d'évaluer l'influence de chaque feature sur les résultats du modèle, facilitant ainsi la sélection ou la suppression des variables.

7 Modèles & Calibration

Les hyperparamètres sont des paramètres qui contrôlent le comportement des modèles et qui ne sont pas appris directement lors de l'entraînement. Leur réglage est crucial pour obtenir des prédictions robustes et précises. Dans notre notebook, nous avons défini pour chaque modèle un ensemble d'hyperparamètres à explorer. Par exemple :

- Pour **Gradient Boosting Survival Analysis**, nous avons optimisé :
  - le nombre d'estimateurs $n_{\text{estimators}}$ (nombre d'arbres),
  - la profondeur maximale des arbres max_depth,
  - le taux d'apprentissage η (learning rate),
  - et d'autres paramètres comme le sous-échantillonnage (subsample) ou la sélection aléatoire des features (max_features).
- Pour le **Random Survival Forest (RSF)**, les hyperparamètres incluent :
  - le nombre d'arbres dans la forêt $n_{\text{estimators}}$,
  - la profondeur maximale max_depth,
  - et éventuellement des paramètres liés à la taille des nœuds (nombre minimum d'échantillons par feuille ou par split).

Afin d'optimiser ces paramètres, nous avons utilisé une *grid search* intégrée dans un schéma de validation croisée. L'idée est de parcourir toutes les combinaisons possibles des hyperparamètres définis et de sélectionner celle qui maximise le C-index, une métrique qui mesure la capacité du modèle à correctement ordonner les risques entre les individus. La procédure se déroule comme suit :

1. Définir une grille d'hyperparamètres pour chaque modèle.
2. Utiliser une validation croisée interne (*inner CV*) pour évaluer chaque combinaison sur une partie des données d'entraînement.
3. Sélectionner la meilleure combinaison en fonction du C-index.
4. Évaluer la performance du modèle optimisé sur un ensemble de test externe via une validation croisée externe (*nested CV*) afin d'obtenir une estimation non biaisée de la performance.

Cette approche permet de réduire le risque de surapprentissage des hyperparamètres et d'obtenir une mesure fiable du pouvoir prédictif du modèle.

8 Conclusion

Nous n'avons pas couvert toutes les tentatives réalisées, chacune ayant ses spécificités, mais plusieurs approches ont été expérimentées. Nous avons notamment essayé des combinaisons de modèles classiques, une approche de Deep Learning ainsi que diverses configurations

Modèle	C-index (%)
CoxPHSurvivalAnalysis	74.46%
GradientBoostingSurvivalAnalysis	76.63%
RandomSurvivalForest	75.54%

Table 10: Performance des différents modèles de survie.

d'ensembles de features et de paramètres. Pour résumer nos résultats, nous avons retenu les performances suivantes :

Les paramètres optimaux utilisés pour ces modèles sont résumés dans le tableau suivant :

Modèle	Paramètres optimaux

GradientBoostingSurvivalAnalysis	`loss`: `'coxph'`
	`max_depth`: 2
	`learning_rate`: 0.05
	`n_estimators`: 335
	`subsample`: 0.55
	`max_features`: `'sqrt'`
	`min_samples_split`: 3
	`random_state`: 26

RandomSurvivalForest	`n_estimators`: 200 `max_depth`: None `min_samples_split`: 50 `min_samples_leaf`: 20 `max_features`: `'sqrt'`

Table 11: Paramètres optimaux pour les modèles de survie retenus.

Nous avons également testé une approche de Deep Learning via `DeepHit`, une méthode proposée dans la littérature pour la prédiction de la survie. Bien que cette approche ait fourni des résultats satisfaisants avec un nombre restreint de features, la complexité de configuration du modèle augmente significativement avec le nombre de variables. Étant donné que les performances obtenues avec nos modèles classiques étaient déjà satisfaisantes, nous n'avons pas poursuivi davantage le développement de l'approche Deep Learning.

Ces différentes méthodes nous ont permis de nous positionner, dès le départ, parmi les premières places du challenge. Le challenge étant toujours en cours et de nombreux participants continuant d'améliorer leurs scores, cela nous a motivés à continuer d'innover pour maintenir un certain classement.