RAPORT DATA CHALLENGE

ENS DATA CHALLENGE — QUBE RESEARCH & TECHNOLOGIES

QRT DATA CHALLENGE
PRÉDICTION DE SURVIE POUR LA LEUCÉMIE MYÉLOÏDE AIGUË

ROBIN GUICHON

03/09/2025 - 15/12/2025

Résumé : *Ce projet développe une pipeline complète pour la prédiction de survie des patients atteints de Leucémie Myéloïde Aiguë (AML) dans le cadre du challenge QRT-Gustave Roussy. À partir d'un jeu de données complexe mêlant variables cliniques tabulaires, mutations génomiques éparses et comptes-rendus cytogénétiques textuels, une stratégie rigoureuse d'ingénierie des fonctionnalités a été mise en œuvre. Celle-ci intègre l'implémentation des règles expertes de l'ELN 2017, le Target Encoding des gènes et l'analyse sémantique (NLP). Pour répondre à la problématique de la censure à droite, des techniques d'expansion temporelle et de correction semi-supervisée ont été appliquées. L'architecture finale repose sur une méthode de Stacking agrégant neuf modèles distincts (dont Histogram Gradient Boosting et MLP). Cette approche a permis d'atteindre un C-Index de 0.7741, surpassant significativement les modèles standards et validant l'apport de l'hybridation entre expertise métier et Machine Learning avancé.*

Keywords : *Stacked Generalization, Histogram Gradient Boosting, Target Encoding, TF-IDF Vectorization, Truncated SVD, Right-Censoring, Discrete-Time Expansion, IPCW C-Index, KNN Imputation, Hyperparameter Optimization*

Table des Matières

Cadre de l'Étude et Analyse du Sujet

I.1 Contexte Médical et Problématique

I.1.1 Contexte Médical et Enjeux Cliniques

La Leucémie Myéloïde Aiguë (AML) est une hémopathie maligne caractérisée par la prolifération incontrôlée de cellules immatures (blastes). C'est une pathologie d'une grande hétérogénéité biologique et pronostique : si certains patients répondent aux standards, d'autres présentent des formes réfractaires ou des rechutes, entraînant une survie très variable.

L'enjeu clinique majeur réside donc dans la stratification du risque pour orienter la stratégie thérapeutique :

- **Pour les patients à faible risque :** L'objectif est de maximiser la survie en minimisant la toxicité. Ils sont orientés vers des chimiothérapies standards et des soins de support.
- **Pour les patients à haut risque :** Le pronostic sombre justifie une approche agressive, notamment la greffe de cellules souches allogénique. Potentiellement curative mais associée à une forte morbidité, elle est réservée aux patients dont le risque lié à la maladie excède celui de la procédure.

Prédire la survie globale dès le diagnostic permet ainsi d'optimiser la balance bénéfice-risque. C'est ici qu'intervient l'analyse de données : en intégrant paramètres cliniques et marqueurs moléculaires complexes, les modèles prédictifs visent à affiner cette stratification bien au-delà des classifications traditionnelles.

I.1.2 Défi : La Stratification du Risque en Oncologie

L'avènement de la médecine de précision offre une opportunité majeure en oncologie : adapter les stratégies thérapeutiques aux profils biologiques individuels. Ce projet, mené dans le cadre du challenge de l'Institut Gustave Roussy, vise à développer un modèle prédictif du risque de décès pour la **Leucémie Myéloïde Aiguë (AML)**. La complexité de ce défi repose sur l'exploitation de données multicentriques (24 centres) et hétérogènes, nécessitant l'unification de variables cliniques tabulaires, de mutations moléculaires complexes et de données textuelles cytogénétiques.

I.1.3 Problématique Mathématique : Censure et Ranking

La difficulté méthodologique centrale est la **censure à droite**, caractéristique des analyses de survie. Pour les patients vivants en fin d'étude, la durée totale de survie est inconnue. La variable cible est donc composite :

- `OS_YEARS` : Le temps observé.
- `OS_STATUS` : L'indicateur de l'événement (1 pour décès, 0 pour censuré).

L'objectif n'est pas de prédire une date de décès exacte, mais d'établir un **classement** fiable des patients du plus au moins risqué, nécessitant des méthodes d'apprentissage adaptées à cette incertitude partielle.

I.2 Caractérisation des Données

Les données analysées dans ce rapport proviennent du challenge organisé par Qube Research Technologies sur la plateforme Data Challenge de l'ENS [1]. Le jeu de données est structuré en deux ensembles distincts : un jeu d'entraînement (`train`) comprenant **3 323 patients** et un jeu de test (`test`) portant sur **1 193 patients**. Chaque patient est identifié par une clé unique (`ID`) permettant de lier trois sources d'informations complémentaires : les données cliniques, les données moléculaires et, pour l'entraînement, la variable cible de survie.

I.2.1 Données Cliniques

Le fichier de données cliniques présente une structure tabulaire avec une ligne par patient. Il regroupe les informations suivantes :

- **ID :** Identifiant unique du patient.
- **CENTER :** Le centre clinique de prise en charge.
- **BM_BLAST :** Pourcentage de blastes dans la moelle osseuse (*Bone marrow blasts*). Les blastes désignent les cellules sanguines anormales.
- **WBC :** Nombre de globules blancs (*White Blood Cell count*), exprimé en Giga/L.
- **ANC :** Nombre absolu de neutrophiles (*Absolute Neutrophil count*), exprimé en Giga/L.
- **MONOCYTES :** Nombre de monocytes, exprimé en Giga/L.
- **HB :** Taux d'hémoglobine, exprimé en g/dL.
- **PLT :** Nombre de plaquettes (*Platelets count*), exprimé en Giga/L.
- **CYTOGENETICS :** Une description textuelle du caryotype observé dans les cellules sanguines du patient. La notation suit la convention **ISCN**. Un caryotype peut être normal (ex: *46,XX* pour les femmes, *46,XY* pour les hommes) ou anormal. Une anomalie fréquente dans les cellules cancéreuses sanguines est par exemple la perte du chromosome 7 (*monosomy 7* ou *-7*), typiquement associée à une maladie à haut risque.

I.2.2 Données Moléculaires

Les données moléculaires sont structurées différemment, avec une ligne par mutation somatique par patient. Les mutations somatiques sont celles trouvées dans les cellules tumorales mais pas dans les autres cellules du corps. Chaque mutation est décrite par les variables suivantes :

- **ID :** Identifiant unique du patient.
- **CHR, START, END :** Position précise de la mutation sur le génome humain.
- **REF, ALT :** Nucléotide de référence et nucléotide alternatif (mutant).
- **GENE :** Le gène affecté par la mutation.
- **PROTEIN_CHANGE :** La conséquence de la mutation sur la protéine exprimée par le gène donné.
- **EFFECT :** Une catégorisation large des conséquences de la mutation sur un gène donné.
- **VAF (Variant Allele Fraction) :** Représente la **proportion** de cellules porteuses de la mutation délétère.

Une difficulté structurelle notable réside dans la séparation de ces informations en deux fichiers de dimensions distinctes. Alors que les données cliniques sont uniques par patient, les données moléculaires sont multiples, créant une disparité de format. Une étape préliminaire indispensable consistera donc à effectuer une jointure relationnelle sur l'identifiant unique (`ID`). L'objectif sera de consolider l'ensemble des descripteurs dans une matrice unifiée, indicée par patient, afin de rendre ces données hétérogènes exploitables par les algorithmes d'apprentissage.

I.2.3 Variables Cibles et Définition de la Survie

Le fichier de données cibles, fourni pour le jeu d'entraînement, m'offre deux informations cruciales pour caractériser le destin du patient :

- **OS_STATUS :** Un indicateur booléen de l'état vital. La valeur **1** signale un décès, tandis que **0** indique que le patient était toujours vivant lors de son dernier suivi médical.
- **OS_YEARS :** Le temps de survie global, mesuré en années à partir de la date du diagnostic.

La difficulté majeure de ce problème réside dans la censure à droite observée dans ces données cibles. Il s'agit d'une contrainte classique en analyse de survie : pour les patients décédés (Status = 1), la durée de survie est exacte. En revanche, pour les patients encore en vie à la dernière consultation (Status = 0), je ne dispose que d'un temps de survie **partiel**. Je sais qu'ils ont survécu **au moins** jusqu'à ce point, mais leur date de décès réelle demeure inconnue.

L'objectif n'est donc pas de prédire une durée exacte, mais d'estimer un score de risque continu. Comme la métrique d'évaluation l'indique, l'échelle absolue de ces prédictions importe peu, seul l'ordre relatif est déterminant. La logique est la suivante : pour deux patients i et j, une prédiction de risque plus faible pour i que pour j ($R_i < R_j$) implique que le modèle estime que le patient i survivra plus longtemps que le patient j.

I.3 Métrique d'Évaluation : Le C-Index

L'évaluation de la performance du modèle ne peut reposer sur des métriques de régression classiques (ex : MSE), car celles-ci pénalisent l'écart absolu entre une prédiction et une réalité. Or, dans mon contexte, la valeur absolue du risque importe peu. C'est la capacité du modèle à discriminer les patients à haut risque de ceux à faible risque qui est primordiale.

La métrique retenue est donc le **C-index** (*Concordance Index*), adapté pour les données censurées via la pondération **IPCW** (*Inverse Probability of Censoring Weighting*).

I.3.1 Le Principe de Concordance

Le C-index évalue la qualité du classement relatif. Le principe repose sur l'analyse de toutes les "paires comparables" de patients (i, j) dans le jeu de données.
Pour une paire donnée où le patient i est décédé à un temps T_i et le patient j à un temps T_j (avec $T_i < T_j$) :

- Le modèle est dit **concordant** s'il a prédit un score de risque plus élevé pour le patient qui a survécu le moins longtemps ($R_i > R_j$).
- Le C-index global est le ratio des paires correctement ordonnées sur le nombre total de paires comparables

$$C = \frac{\text{Nombre de Paires Concordantes}}{\text{Nombre Total de Paires Comparables}} \tag{3.1}$$

Un score de **0.5** équivaut à une prédiction aléatoire (aucun pouvoir discriminant), tandis qu'un score de **1** indique une concordance parfaite.

I.3.2 Gestion de la Censure et Pondération IPCW

L'application directe du C-index est mise en échec par la **censure à droite**. En effet, si un patient j est censuré à un instant T_j (perdu de vue ou vivant à la fin de l'étude) et qu'un patient i est vivant à un temps $T_i < T_j$, la paire (i, j) devient incomparable : j'ignore si i survivra finalement plus ou moins longtemps que j. Ignorer ces paires ou les traiter naïvement introduirait un biais statistique majeur dans l'évaluation.

Pour corriger ce biais, le challenge utilise la variante **IPCW**. Cette méthode étend le C-index traditionnel en appliquant des poids spécifiques aux paires de données. L'approche consiste à pondérer chaque observation en fonction de la probabilité inverse qu'elle ait été observée (non censurée). Concrètement, les patients qui ne sont pas censurés représentent statistiquement ceux qui ont été censurés auparavant et qui présentaient des caractéristiques similaires. Cette re-pondération permet de reconstruire artificiellement la population complète et de calculer un score de concordance non biaisé sur l'horizon temporel de l'étude (tronqué à 7 ans dans le benchmark).

Stratégie de Traitement des Données

Le **Feature Engineering** constitue un des pilier de mon approche. Les données brutes, en particulier les listes de mutations génétiques, ne sont pas directement exploitables par des modèles d'apprentissage automatique classiques. Ma stratégie s'est articulée autour de trois axes : la structuration de l'information génétique fragmentée, l'extraction sémantique des données textuelles cytogénétiques, et l'enrichissement des données par des méthodes non supervisées. Je détaille ci-dessous chaque étape de ce processus.

II.1 Traitement des Données Moléculaires

Les données moléculaires se présentent initialement sous une forme relationnelle complexe : une liste de mutations (une ligne par mutation) où chaque patient peut posséder un nombre variable d'altérations génétiques, ou aucune. Pour transformer cette information en un vecteur de caractéristiques de taille fixe par patient, j'ai

mis en place deux techniques complémentaires : la vectorisation des fréquences alléliques et l'encodage ciblé du risque (*Target Encoding*).

II.1.1 Construction de la Matrice VAF (Pivot)

La première étape a consisté à capturer l'intensité de la présence des mutations les plus fréquentes. Plutôt que de traiter l'ensemble des gènes, ce qui aurait créé une matrice trop creuse, j'ai sélectionné les **50 gènes les plus récurrents** dans la cohorte (ex : *NPM1*, *FLT3*, *DNMT3A*).
J'ai ensuite pivoté les données pour créer une matrice où :

- Chaque ligne correspond à un patient (ID).
- Chaque colonne correspond à l'un des 50 gènes sélectionnés.
- La valeur correspond à la **VAF** maximale observée pour ce gène chez ce patient.

Les patients ne présentant pas de mutation pour un gène donné se voient attribuer une valeur de 0. Cette transformation permet de traduire les mutations en un score chiffré directement utilisable par le modèle. En complément, une variable `mut_count` a été créée pour dénombrer simplement la charge mutationnelle totale de chaque patient.

II.1.2 Target Encoding : Quantification du Risque Génétique

Au-delà de la présence ou de la fréquence d'une mutation, l'information cruciale réside dans son impact pronostique (favorable ou défavorable). Pour capturer cet effet sans augmenter la dimensionnalité, j'ai appliqué une technique de **Target Encoding** sur la variable `GENE`. Une description détaillée de cette méthode de prétraitement pour les variables à haute cardinalité a été réalisée par Micci-Barreca **[2]**.

L'approche consiste à remplacer le nom catégoriel de chaque gène par une valeur numérique représentative du risque associé, calculée sur l'ensemble d'entraînement :

1. Pour chaque gène, je calcule la **médiane de survie globale** (`OS_YEARS`) des patients porteurs de cette mutation.
2. Cette valeur est mappée à chaque mutation dans le jeu de données. Pour les gènes rares non vus dans l'entraînement, la médiane globale de la population est utilisée comme valeur par défaut.

Enfin, ces scores sont agrégés au niveau du patient pour créer deux nouvelles variables synthétiques fortes :

- `worst_gene` : Le score de risque le plus élevé parmi toutes les mutations du patient correspondant à la mutation la plus "dangereuse" ou au pronostic le plus sombre.
- `mean_gene` : La moyenne des scores de risque de toutes ses mutations, reflétant le profil de risque génétique moyen.

Cette méthode permet d'injecter directement une connaissance statistique du risque biologique dans les modèles, focalisant ainsi le modèle sur les signaux prédictifs les plus forts.

II.2 Traitement des Données Cliniques et Textuelles (NLP)

Les données cliniques constituent le second pilier de mon modèle. Si certaines variables sont numériques et directement exploitables (âge, numération sanguine), la variable `CYTOGENETICS` présente un défi majeur : il s'agit d'une chaîne de caractères décrivant des anomalies chromosomiques complexes. Pour extraire le maximum d'information de ce champ textuel, j'ai combiné une approche experte (basée sur des règles médicales) et une approche statistique non supervisée.

II.2.1 Analyse Sémantique de la Cytogénétique

Le caryotype est un facteur pronostique déterminant dans la leucémie myéloïde aiguë. Ma stratégie d'extraction s'est décomposée en trois niveaux :

1. Approche Experte : Classification ELN par Regex

J'ai implémenté un système de règles basées sur des expressions régulières (*Regex*) pour détecter des motifs spécifiques associés à la classification ELN (European LeukemiaNet voir **Annexe A**). Cette variable catégorielle `eln` encode le risque en trois niveaux :

- **Favorable Score 1** : Détection de motifs tels que `inv(16)`, `t(8;21)` ou `t(16;16)`.
- **Adverse Score 3** : Détection d'anomalies à haut risque comme la monosomie 7 (`-7`), la délétion 5q (`del(5)`) ou des caryotypes complexes (`complex`).
- **Intermédiaire Score 2** : Tous les autres cas.

Cette variable `eln` est ensuite croisée avec la charge tumorale pour créer une variable d'interaction forte : `int_eln_load`.

2. Approche Statistique : TF-IDF et SVD

Pour capturer des nuances plus subtiles non couvertes par les règles expertes, j'ai traité le champ `CYTOGENETICS` comme un document textuel **[3]**.

- **Vectorisation TF-IDF :** J'ai utilisé une vectorisation au niveau des caractères (n-grams de 3 à 5 caractères) pour être robuste aux variations d'écriture.
- **Réduction de Dimension (SVD) :** La matrice résultante étant de haute dimension, j'ai appliqué une *Truncated SVD* (décomposition en valeurs singulières) pour projeter l'information textuelle sur **12 composantes latentes** (`svd_0` à `svd_11`). Ces vecteurs capturent la variance structurelle des descriptions chromosomiques.

3. Target Encoding Cytogénétique

De manière analogue aux gènes, j'ai calculé la médiane de survie associée à chaque description cytogénétique exacte (`cyto_risk`). Pour éviter le sur-apprentissage sur des descriptions rares, seules les occurrences apparaissant au moins 10 fois ont été encodées spécifiquement ; les autres ont reçu la médiane globale.

II.2.2 Transformation et Normalisation des Biomarqueurs

L'analyse exploratoire des variables sanguines (`WBC`, `PLT`, `BM_BLAST`) a révélé des distributions fortement asymétriques avec une queue lourde à droite. Pour stabiliser la variance et améliorer la convergence des modèles linéaires et neuronaux, j'ai appliqué une transformation logarithmique :

$$x' = \log(1 + x) \tag{2.1}$$

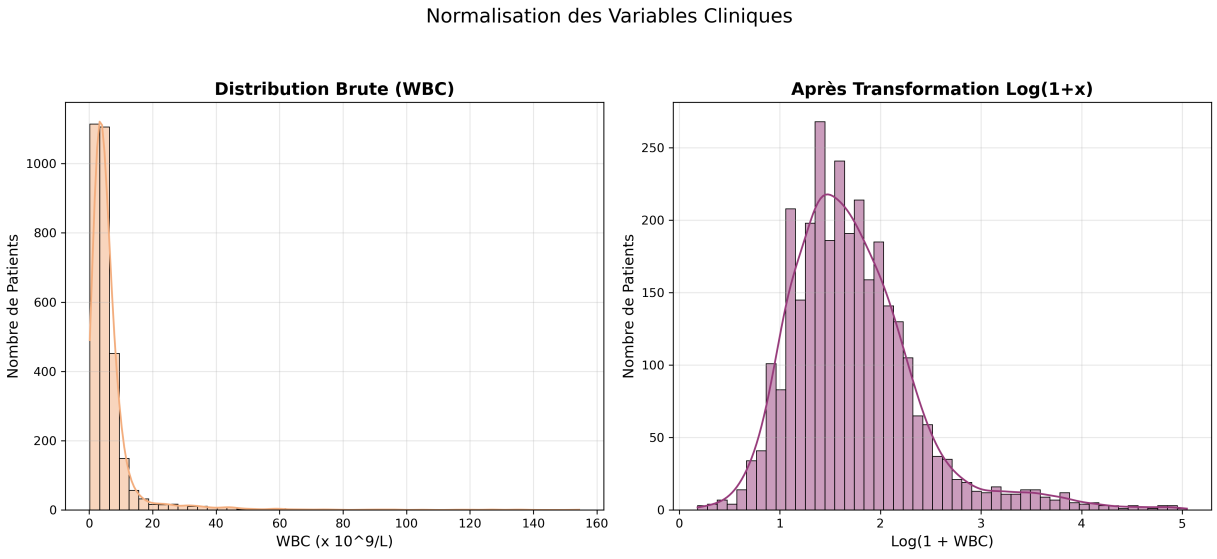

Figure II.2.1 – Impact de la transformation logarithmique sur la distribution de la variable WBC

La figure ci-dessus illustre l'effet de la transformation $x' = \log(1 + x)$ sur la variable WBC (White Blood Cells). À gauche, la distribution brute est fortement asymétrique, dominée par quelques valeurs extrêmes qui écrasent la majorité des patients dans les faibles valeurs. À droite, après transformation, la distribution se déplie et se rapproche d'une gaussienne. Cette normalisation est cruciale pour permettre aux modèles (réseaux de neurones, régressions linéaires) de capter efficacement les variations fines entre les patients, sans être biaisés par les valeurs aberrantes.

En complément, j'ai créé des variables synthétiques métier, notamment la charge leucémique totale (`load`), définie comme le produit du taux de blastes par le nombre de globules blancs. L'ensemble des variables continues a ensuite été imputé par la médiane avant d'être normalisé.

II.3 Apprentissage Non-Supervisé et Imputation

Après avoir traité les données moléculaires et cliniques séparément, j'ai cherché à capturer des structures latentes globales au sein de la cohorte. Pour cela, j'ai eu recours à des méthodes d'apprentissage non supervisé avant de finaliser la préparation du jeu de données par une imputation multivariée.

II.3.1 Segmentation des Patients par Clustering (K-Means)

L'hypothèse sous-jacente est qu'il existe des sous-groupes de patients partageant des profils biologiques similaires qui ne sont pas explicitement capturés par les variables individuelles.
Pour révéler ces structures, j'ai appliqué l'algorithme des K-Means sur un sous-ensemble de caractéristiques biologiques composé :

- Des fréquences alléliques des mutations (`vaf_*`).
- Des composantes sémantiques issues de la cytogénétique (`svd_*`).

J'ai fixé le nombre de groupes à $k = 12$. Chaque patient s'est vu attribuer un cluster d'appartenance. Cette information catégorielle a ensuite été transformée en variables binaires (*One-Hot Encoding*) et ajoutée au jeu de données. Cette étape de Feature Augmentation permet aux modèles supervisés ultérieurs de bénéficier d'une information synthétique sur le profil type du patient.

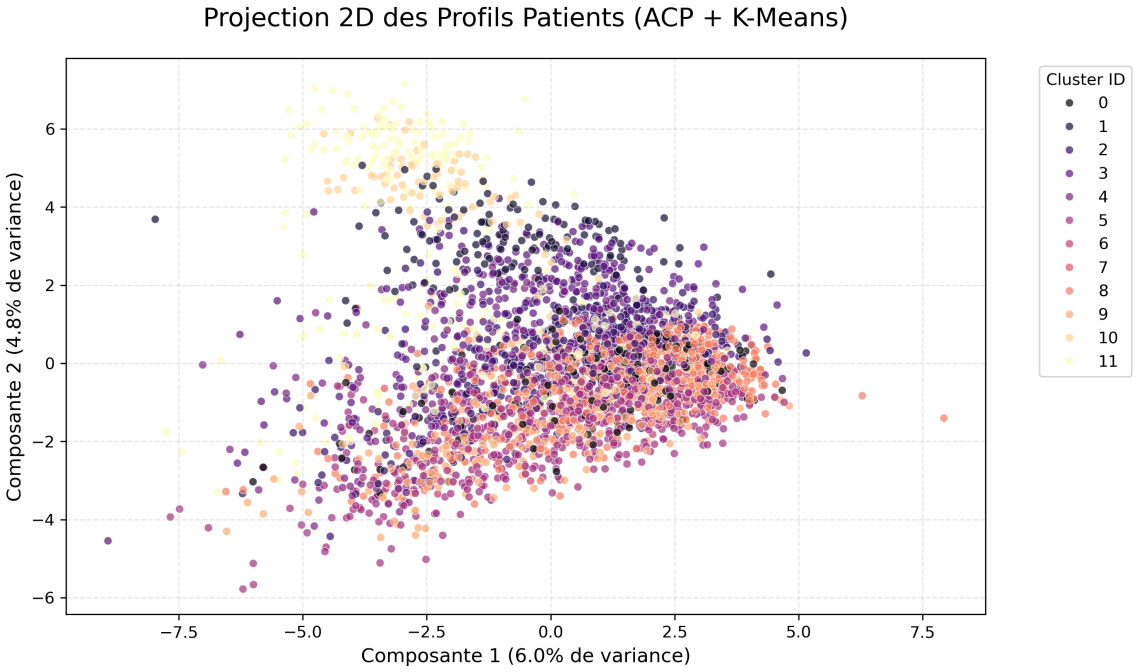

Figure II.3.1 – Projection 2D des profils patients sur les axes principaux de l'ACP

La figure ci-dessus permet de vérifier la pertinence de la segmentation des patients. Mes données comportant initialement plus de 80 dimensions, j'ai utilisé une Analyse en Composantes Principales (ACP) pour projeter les patients sur un plan 2D.

On observe que la population ne forme pas un nuage uniforme et aléatoire, mais s'organise en structures distinctes. Le fait que les points de même couleur (appartenant au même cluster K-Means) se regroupent spatialement confirme que l'algorithme a réussi à identifier des profils biologiques homogènes. Cette segmentation permet de capturer des sous-types de la maladie que les modèles pourront exploiter via la variable `cluster_id`.

II.3.2 Stratégie d'Imputation par Plus Proches Voisins (KNN)

Bien que certaines valeurs manquantes aient été traitées sommairement lors des étapes précédentes (par la médiane), une imputation plus fine est nécessaire pour maximiser la cohérence des données avant l'entraînement. Plutôt que d'utiliser des statistiques globales (moyenne/médiane) qui écrasent la variance et ignorent les corrélations entre variables, j'ai opté pour le KNNImputer (*K-Nearest Neighbors Imputation*).

— **Principe :** Pour chaque valeur manquante, l'algorithme identifie les $k = 15$ patients les plus proches dans l'espace multidimensionnel des caractéristiques.
— **Calcul :** La valeur imputée est une moyenne pondérée des valeurs de ces voisins.

Cette approche préserve la structure locale des données : on estime la valeur manquante d'un patient en se basant sur ceux qui lui ressemblent cliniquement et génétiquement.

À l'issue de cette phase, je dispose d'une matrice de données complète, enrichie de clusters et de variables textuelles/génétiques transformées, prête à être soumise aux algorithmes de prédiction de survie.

II.4 La Gestion Avancée de la Censure

La spécificité majeure de ce problème réside dans la présence de données censurées à droite (patients encore vivants à la fin de l'étude). Ignorer cette information ou supprimer ces patients introduirait un biais massif. Pour exploiter l'intégralité de la cohorte avec des algorithmes standards (qui ne gèrent pas nativement la censure comme le modèle de Cox), j'ai développé deux stratégies de transformation de la variable cible.

II.4.1 Approche 1 : Expansion Temporelle

Pour mes modèles de classification (ex : *Gradient Boosting Classifier*), j'ai transformé le problème de régression de survie en une suite de problèmes de classification binaire séquentielle. Cette méthode est décrite dans les recherches d'Allison **[4]**.
J'ai discrétisé le temps en intervalles de 6 mois ($t \in \{0, 0.5, 1.0, \ldots, 10\}$ ans). Pour chaque patient i, j'ai dupliqué ses données pour chaque intervalle t_k où il est "à risque" (vivant et observé).

— Si le patient est vivant au temps t_k, la cible pour cet intervalle est $y_{i,k} = 0$.
— Si le patient décède dans l'intervalle $[t_k, t_{k+1}[$, la cible est $y_{i,k} = 1$ et l'expansion s'arrête.

Cette technique, implémentée dans la fonction `expand`, multiplie la taille du jeu de données mais permet au modèle d'apprendre la probabilité conditionnelle de décès à chaque étape, mimant ainsi la dynamique temporelle du risque.

II.4.2 Approche 2 : Correction Semi-Supervisée de la Cible

Pour mes modèles de régression (Random Forest, MLP, KNN, Lasso), qui prédisent un score continu, je ne peux pas utiliser directement le temps de survie observé (`OS_YEARS`) pour les patients censurés, car il sous-estime leur véritable survie.
J'ai donc mis en place une stratégie de correction semi-supervisée via la fonction `get_log_y` :

1. **Entraînement auxiliaire :** Un modèle *Random Forest* intermédiaire est entraîné exclusivement sur les patients non-censurés (décédés) pour apprendre la relation entre les caractéristiques biologiques et la durée de survie réelle.

2. **Prédiction latente :** Ce modèle prédit une durée de survie hypothétique $\hat{T}_i$ pour les patients censurés.
3. **Correction :** Pour chaque patient censuré observé jusqu'au temps C_i, je définis une nouvelle cible corrigée Y'_i :

$$Y'_i = \max(C_i, \hat{T}_i) \tag{4.1}$$

Cette approche garantit que la cible utilisée pour l'entraînement est au moins égale au temps de suivi observé, tout en complétant l'information manquante par une estimation statistiquement plausible basée sur le profil du patient. La cible finale est ensuite passée au logarithme ($\log(1 + Y)$) pour stabiliser la variance de l'erreur.

Grâce à ces transformations, j'ai converti un problème complexe de survie censurée en tâches d'apprentissage supervisé classiques, me permettant de déployer un large éventail d'algorithmes performants dans la phase suivante.

Architecture de Modélisation et Résultats

Après avoir transformé mes données brutes en un ensemble cohérent de variables explicatives et traité la problématique de la censure, j'entre dans la phase de modélisation. Cette partie décrit l'architecture complète de ma solution, qui ne repose pas sur un modèle unique, mais sur la combinaison stratégique de plusieurs familles d'algorithmes (Stacking). Je détaillerai d'abord l'ensemble des modèles sélectionnés, puis le processus d'optimisation de leurs hyperparamètres, avant d'exposer mon protocole de validation croisée et les résultats finaux.

Il convient toutefois de préciser que ma démarche s'est initialement appuyée sur les méthodes proposées dans le benchmark, incluant notamment le modèle de Cox standard et LightGBM. Ces méthodes ont servi de première approche avant le développement de mon architecture de Stacking plus complexe. Une description détaillée de ces méthodes de référence est fournie en **Annexe B**.

III.1 Présentation des Algorithmes

Pour capturer la complexité des données hétérogènes et maximiser la robustesse des prédictions, je n'ai pas reposé ma solution sur un algorithme unique. Mon architecture finale est un **Stacking** combinant les prédictions de **9 modèles distincts** issus de 5 familles algorithmiques. Les détails techniques, les hyperparamètres et les fondements mathématiques de chaque algorithme sont exposés en **Annexe C**.

Cette stratégie de diversité me permet de corriger les biais individuels de chaque modèle grâce à leur complémentarité :

III.1.1 Le Cœur du Système : Histogram Gradient Boosting (4 modèles)

L'algorithme HGBT constitue la colonne vertébrale de mon architecture (poids total de 50% dans le stacking) en raison de sa rapidité et de sa gestion native des valeurs manquantes **[5][6]**. J'ai déployé 4 variantes pour capturer différentes facettes du risque :

— **1 Classifieur (Log-Loss) :** Entraîné sur les données étendues temporellement, il modélise la dynamique du risque instantané de décès au cours du temps.
— **3 Régresseurs (Poisson, Gamma, MSE) :** Ces modèles prédisent directement le score de risque. L'utilisation de fonctions de perte distinctes permet de mieux gérer les distributions atypiques (Poisson pour les événements rares/haut risque, Gamma pour l'asymétrie des durées de survie).

III.1.2 Le Stabilisateur : Forêts Aléatoires (2 modèles)

Là où le Boosting cherche la performance pure (risque d'overfitting), les Random Forests apportent de la stabilité grâce au principe du Bagging (moyenne d'arbres décorrélés). J'ai utilisé **2 variantes** :

— **RF Standard :** Pour capturer les tendances générales robustes.
— **RF Profond :** Autorisé à descendre plus profondément dans les arbres pour saisir des interactions plus fines entre les variables.

III.1.3 L'Abstracteur : Perceptron Multi-Couches (1 modèle)

Un réseau de neurones profond a été intégré pour sa capacité à modéliser des **relations non-linéaires complexes** que les modèles à base d'arbres peinent parfois à saisir. Il joue le rôle de spécialiste des interactions latentes entre le profil génétique (VAF) et les données cliniques, apportant une vision différente des données.

III.1.4 La Diversité Locale et Linéaire : KNN et Lasso (2 modèles)

Enfin, deux modèles plus simples agissent comme des gardes-fous :

— **K-Nearest Neighbors (KNN) :** Il base sa prédiction sur la similarité locale plutôt que sur des règles, lissant les résultats.
— **Régression Lasso :** Ce modèle linéaire régularisé capture les signaux simples et forts, garantissant que l'ensemble ne rate pas les évidences cliniques.

L'agrégation finale de ces 9 modèles se fait par une moyenne pondérée de leurs rangs, exploitant la faible corrélation entre ces différentes familles pour réduire la variance globale de l'erreur.

III.2 Optimisation des Hyperparamètres

La performance d'un modèle d'apprentissage dépend fortement de ses hyperparamètres, c'est-à-dire des configurations fixées **avant** l'entraînement. Pour maximiser la performance de mes modèles clés, notamment les Gradient Boosting, j'ai mis en place une procédure d'optimisation rigoureuse en amont du Stacking final.

III.2.1 Méthodologie : Recherche sur Grille (GridSearch)

J'ai utilisé la méthode Grid Search avec validation croisée. Cette approche consiste à tester de manière exhaustive toutes les combinaisons possibles d'une grille de paramètres prédéfinie pour identifier la configuration optimale. Pour chaque combinaison de paramètres, le modèle est évalué par validation croisée (ici sur 3 plis) :

— Pour le classifieur (`gb1`), j'ai maximisé l'aire sous la courbe ROC (`roc_auc`).
— Pour les régresseurs (`gb2`, `gb3`, `gb4`), j'ai minimisé l'erreur quadratique moyenne négative (`neg_mean_squared_error`).

III.2.2 Paramètres Optimisés pour le Gradient Boosting

Le *Histogram Gradient Boosting* est particulièrement sensible à certains hyperparamètres contrôlant la complexité et la régularisation. J'ai fait varier les paramètres suivants :

— **Taux d'apprentissage (`learning_rate`) :** Contrôle la contribution de chaque arbre à la prédiction finale. J'ai testé une plage de valeurs faibles (de 0.01 à 0.1) pour favoriser une convergence lente mais précise.
— **Profondeur des arbres (`max_depth`) :** Limite la complexité de chaque arbre individuel. J'ai exploré des profondeurs allant de 2 à 9. Une profondeur faible permet d'éviter le sur-apprentissage, agissant comme une régularisation structurelle.
— **Régularisation L2 (`l2_regularization`) :** Ajoute une pénalité sur les feuilles de l'arbre pour lisser les prédictions. J'ai testé des valeurs entre 0.5 et 1.5.
— **Nombre d'itérations (`max_iter`) :** Le nombre total d'arbres construits (testé entre 150 et 200).

Cette étape d'optimisation m'a permis de fixer des configurations robustes, garantissant que chaque brique de mon ensemble final fonctionne à son plein potentiel avant d'être utilisée dans le stacking.

III.3 Stratégie d'Agrégation et Inférence (Stacking)

La dernière étape de mon pipeline consiste à fusionner les prédictions de mes 9 modèles pour produire un score de risque unique et robuste. Pour l'architecture finale, j'ai opté pour une stratégie d'agrégation de modèles (Stacking), une méthode de généralisation théorisée par Wolpert **[7]**, afin de réduire la variance des prédictions.

III.3.1 Le Stacking par Moyenne Pondérée

L'agrégation repose sur une combinaison linéaire des rangs prédits par chaque modèle. Les poids ont été déterminés empiriquement en se basant sur les performances individuelles observées lors de la validation croisée, tout en maintenant une diversité algorithmique.
La formule du score final S_{final} pour un patient x est :

$$S_{final}(x) = \sum_{m=1}^{9} \omega_m \cdot \text{rang}(\hat{y}_m(x)) \tag{3.1}$$

La distribution des poids ω reflète ma confiance dans chaque famille d'algorithmes :

- **Gradient Boosting (Total : 50%)** : C'est le cœur du système. Le classifieur temporel (`gb1`) reçoit le poids individuel le plus fort ($\omega = 0.25$) car il modélise le mieux la dynamique de survie. Les régresseurs (Poisson, Gamma, MSE) complètent ce signal ($\omega = 0.25$ au total).
- **Réseau de Neurones (MLP : 15%)** : Un poids significatif est accordé au MLP pour sa capacité à capter les non-linéarités globales que les arbres manquent.
- **Forêts Aléatoires (Total : 20%)** : Les deux variantes (profonde et standard) reçoivent chacune 10%, agissant comme un filet de sécurité pour stabiliser les prédictions.
- **KNN (10%) et Lasso (5%)** : Ces modèles ont des poids plus faibles mais essentiels pour apporter une perspective locale (KNN) et linéaire (Lasso), corrigeant les biais des méthodes plus complexes.

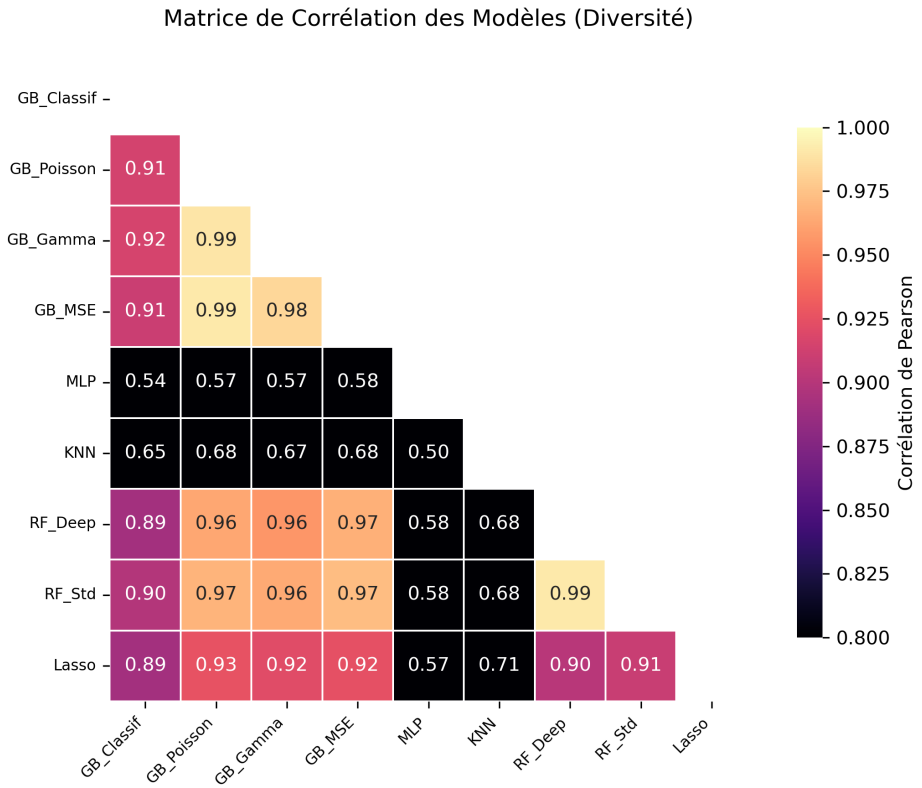

Figure III.3.1 – Matrice de corrélation de Pearson entre les prédictions des modèles de base (Jeu de validation)

La matrice ci-dessus mesure la similarité des prédictions entre mes différents algorithmes. Pour qu'un Stacking

soit performant, il faut combiner des modèles qui sont bons, mais qui ne font pas les mêmes erreurs au même moment c'est le principe de diversité.

— **Zones claires (Corrélation > 0.95) :** On observe une forte ressemblance entre les modèles de la même famille (les différentes variantes de Gradient Boosting sont très proches).
— **Zones sombres (Corrélation < 0.90) :** C'est le point clé. Le KNN (basé sur la distance) et le Lasso (modèle linéaire) montrent une corrélation plus faible avec les modèles d'arbres.

Cette diversité structurelle prouve que ces modèles apportent une information complémentaire unique, permettant à l'ensemble final d'être plus robuste que le meilleur modèle individuel.

III.3.2 Ré-entraînement Final et Génération des Prédictions

Une fois l'architecture et les hyperparamètres validés par la procédure de Cross-Validation, il est crucial d'exploiter l'intégralité des données disponibles pour l'inférence finale.
J'ai donc procédé à un ré-entraînement complet de tous les modèles sur l'ensemble du jeu d'entraînement (3323 patients), sans retenue de validation :

1. **Préparation Globale :** La transformation temporelle (`expand`) et la correction de la cible (`get_log_y`) sont ré-appliquées sur la totalité des données d'entraînement.
2. **Fit Final :** Les 9 modèles sont entraînés sur ces données maximisées.
3. **Inférence :** Les prédictions sont générées pour les 1193 patients du jeu de test (`X_test`), puis transformées en rangs et agrégées selon les poids définis.

Le résultat est un vecteur de scores de risque continus (`risk_score`), prêt pour la soumission, qui capitalise sur toute l'information statistique disponible.

III.4 Analyse des Résultats

L'évaluation finale de mon approche a été réalisée via la soumission de mes prédictions sur la plateforme du Data Challenge, permettant de confronter mon modèle à des données de test inédites.

III.4.1 Performance sur le Leaderboard

À l'issue de mon processus de modélisation, l'architecture de Stacking complète, entraînée sur l'intégralité des données disponibles, a permis d'atteindre un score **C-Index de 0.7640** sur le classement public.
Ce résultat valide la robustesse de mon approche et surpasse significativement le benchmark initial (0.6541).

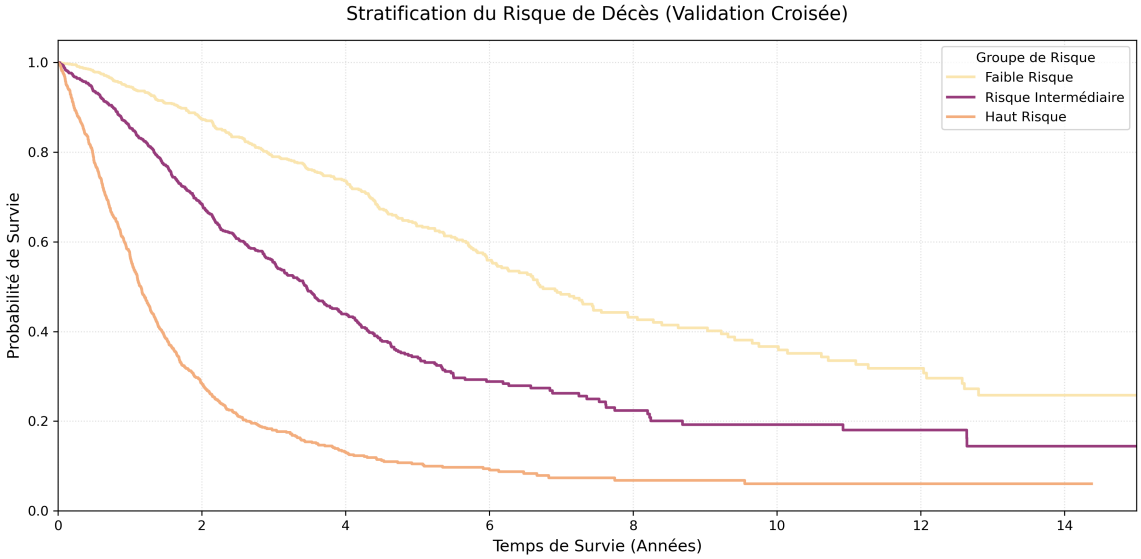

Figure III.4.1 – Courbes de survie Kaplan-Meier par groupe de risque prédit (Validation)

La figure ci-dessus démontre la capacité opérationnelle du modèle à trier les patients selon leur gravité. J'ai divisé la cohorte de validation en trois tiers (Faible, Intermédiaire, Haut Risque) basés sur les scores de mon algorithme.

- Groupe Faible Risque (Beige) : La courbe reste élevée et stable (survie > 80% à long terme). Le modèle identifie avec succès les patients de bon pronostic.
- Groupe Haut Risque (Orange) : La courbe chute brutalement dès les premières années. Le modèle a réussi à isoler les patients nécessitant une intervention thérapeutique urgente et agressive.
- **Discrimination :** L'absence de chevauchement entre les courbes (notamment avec le groupe Intermédiaire en violet) et leur écartement net valident la robustesse du score. Le modèle ne se contente pas de prédire une moyenne, il sépare efficacement les populations.

III.4.2 L'Apport Décisif du Stacking et du Feature Engineering

Ce score de 0.7640 est l'aboutissement d'une progression itérative. Lors de mes premières expérimentations, l'utilisation de modèles de survie isolés, tels que les *Random Survival Forests* (RSF) ou le *Survival Lasso*, montrait des limites de performance. Malgré leurs qualités intrinsèques, ces modèles individuels plafonnaient systématiquement en dessous de la barre des 0.75.
Le franchissement de ce seuil de performance repose sur la synergie de deux facteurs clés :

1. **La puissance de l'Ensemble Learning :** La combinaison de modèles aux biais complémentaires : Gradient Boosting pour la performance, Random Forest pour la stabilité, MLP pour la non-linéarité, a permis de réduire la variance globale de l'erreur et de généraliser bien mieux que n'importe quel modèle unique.
2. **La richesse du Feature Engineering :** L'optimisation des algorithmes n'aurait pas suffi sans un travail approfondi sur les données. La structuration des mutations génétiques (Target Encoding) et l'extraction d'information depuis le texte cytogénétique (NLP/SVD) ont fourni aux modèles des signaux prédictifs de haute qualité, essentiels pour discriminer les patients aux profils complexes.

Conclusion

Au terme de ce Data Challenge proposé par Qube Research Technologies en partenariat avec l'Institut Gustave Roussy, j'ai pu mesurer l'apport crucial de la science des données pour la stratification du risque dans la Leucémie Myéloïde Aiguë (AML). Ce projet a constitué une véritable passerelle entre ma formation académique et la réalité de la recherche opérationnelle, m'obligeant à dépasser l'application standard du Machine Learning.

Ce travail a été l'occasion de **consolider et d'appliquer concrètement** les modèles théoriques étudiés en cours, tels que les Forêts Aléatoires, les Perceptrons Multi-Couches (MLP) et les méthodes de Clustering (K-Means, ACP). Cependant, la complexité des données (hétérogènes et censurées) m'a poussé à acquérir, en autonomie, la maîtrise de **nouvelles techniques avancées** non couvertes par le programme standard. L'implémentation du *Gradient Boosting* sur histogrammes, l'ingénierie textuelle (NLP) et surtout la gestion mathématique de la censure (via l'Expansion Temporelle) ont été déterminantes.

Cette alliance entre les fondamentaux académiques et l'exploration de nouveaux algorithmes a fondé mon architecture de **Stacking**. Celle-ci tire parti de la complémentarité entre la stabilité des méthodes classiques et la précision des modèles de Boosting nouvellement maîtrisés. La qualité de la représentation des données, obtenue par extraction sémantique et *Target Encoding*, a transformé des signaux bruts en prédicteurs biologiques forts.

Les résultats valident la pertinence de cette démarche d'apprentissage continu : avec un **Score de 0.7640**, ma solution surpasse significativement le benchmark (0.6541) et les modèles de survie traditionnels. Ce gain de performance traduit une capacité réelle à mieux discriminer les patients à haut risque, illustrant comment l'hybridation entre socle théorique solide et agilité technique permet de répondre aux défis complexes de la médecine de précision.

Annexes

Annexe A : Implémentation de la Stratification du Risque ELN 2017

La stratification du risque établie par l'European LeukemiaNet (ELN) est le standard international actuel pour le pronostic de la Leucémie Myéloïde Aiguë (AML). Publiée en 2017 par Döhner et al. dans la revue Blood **[8]**, cette classification synthétise des décennies de recherche clinique pour diviser les patients en trois groupes de pronostic distincts : Favorable, Intermédiaire et Défavorable.

Principe Général

Cette classification repose sur une approche hiérarchique combinant deux types d'informations biologiques :

1. **La Cytogénétique :** L'analyse des anomalies chromosomiques à grande échelle (translocations, inversions, pertes de chromosomes).
2. **La Biologie Moléculaire :** La détection de mutations sur des gènes spécifiques (tels que NPM1, FLT3, TP53).

Utilité Clinique : En pratique médicale, l'ELN ne sert pas uniquement à prédire la survie, mais constitue un véritable outil d'aide à la décision thérapeutique. Elle permet d'adapter le traitement à l'agressivité biologique de la maladie :

— Les patients à risque favorable sont généralement traités par chimiothérapie intensive seule, car ils répondent bien au traitement.
— Les patients à risque défavorable sont quasi systématiquement orientés vers une allogreffe de cellules souches hématopoïétiques (greffe de moelle) en première rémission, seule option curative à long terme malgré sa lourdeur.

Cette classification représente une connaissance experte cruciale : elle capture les interactions non-linéaires majeures entre les gènes qui déterminent la survie du patient.

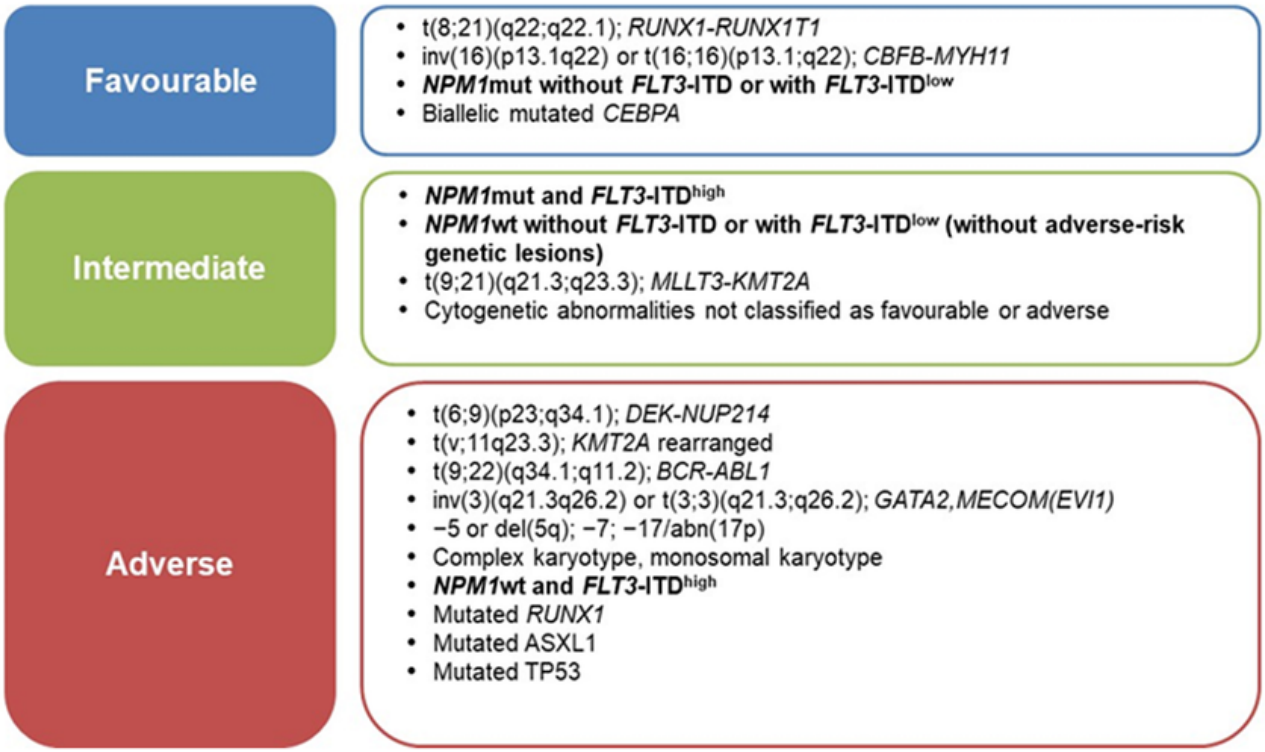

Figure V.1.1 – Tableau de référence de la stratification du risque ELN 2017 (Source : Döhner et al., Blood 2017 **[8]**)

Annexe B : Description et efficacité des modèles du Benchmark

Afin de guider les participants et d'établir une ligne de base pour la compétition, les organisateurs du challenge QRT ont fourni deux approches distinctes. Ces modèles permettent non seulement de tester la chaîne de traitement des données, mais aussi de définir un seuil de performance minimal à dépasser.
Si le premier modèle (LightGBM) est fourni principalement à titre d'exemple technique, c'est le second (modèle de Cox) qui fait office de référence officielle pour le classement académique.

V.2.1 Light Gradient-Boosting Machine

Le premier modèle du benchmark proposé repose sur l'algorithme **Light Gradient-Boosting Machine (LightGBM)**, une méthode d'ensemble fondée sur l'agrégation séquentielle d'arbres de décision faibles dans le cadre du gradient boosting. L'objectif de cette approche est d'approximer une fonction inconnue $f(x)$ en construisant un modèle additif :

$$\hat{f}(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{2.1}$$

où $h_m(x)$ représente le m-ième arbre de décision et γ_m son poids associé. L'apprentissage consiste à minimiser une fonction de perte différentiable $L(y, \hat{f}(x))$ en ajoutant itérativement un nouvel arbre h_m qui approxime le gradient négatif de la perte par rapport aux prédictions précédentes :

$$h_m(x) \approx -\nabla_{\hat{f}_{m-1}} L(y, \hat{f}_{m-1}(x)) \tag{2.2}$$

Ainsi, chaque nouvel arbre vise à corriger les erreurs résiduelles des itérations précédentes. LightGBM se distingue des implémentations classiques du gradient boosting par deux innovations principales :

1. **GOSS (Gradient-based One-Side Sampling) :** Cette technique permet de conserver en priorité les observations présentant un gradient élevé (forte erreur), tout en ne retenant qu'un échantillon contrôlé des gradients faibles. Cela permet d'accélérer l'optimisation tout en préservant la précision du signal d'apprentissage.
2. **EFB (Exclusive Feature Bundling) :** Cette méthode regroupe les variables rarement actives simultanément (données éparses) en un nombre réduit de dimensions, diminuant ainsi la complexité du modèle sans perte d'information significative.

D'un point de vue pratique, LightGBM est particulièrement adapté aux données tabulaires hétérogènes et à forte dimensionnalité, comme les données génomiques. Cependant, cette flexibilité s'accompagne d'un risque accru de sur-apprentissage dans des contextes où la taille d'échantillon reste modérée, comme c'est le cas ici avec environ 3000 patients.

V.2.2 Modèle de risques proportionnels de Cox

Le modèle de Cox constitue la référence standard en analyse de survie et sert de benchmark officiel dans ce challenge. Il s'agit d'un modèle de régression semi-paramétrique visant à estimer le risque instantané de survenue de l'événement (ici, le décès) en fonction d'un vecteur de covariables x.
La fonction de risque conditionnelle est définie comme :

$$\lambda(t|x) = \lambda_0(t) \exp(\beta^T x) \tag{2.3}$$

où $\lambda_0(t)$ représente le risque de base (non paramétré), et β un vecteur de coefficients décrivant l'effet multiplicatif des covariables sur le risque relatif. Cette formulation repose sur l'hypothèse des **risques proportionnels** : les rapports de risque entre deux individus restent constants dans le temps.

Estimation des coefficients Pour l'estimation des coefficients, le modèle de Cox ne cherche pas à paramétrer explicitement le risque de base $\lambda_0(t)$. Au lieu de cela, Cox introduit la **vraisemblance partielle**, qui permet d'estimer les effets des covariables β en utilisant uniquement l'ordre des temps d'événements observés, sans hypothèse sur la forme de $\lambda_0(t)$.
Considérons un instant t_i où l'événement (parmi les m décès) survient pour le patient i. On note $R(t_i)$ l'ensemble des individus encore à risque à cet instant. Dans ce cadre, la probabilité conditionnelle que le patient i, parmi tous les individus dans $R(t_i)$, soit celui qui subit l'événement, s'écrit :

$$P(i|R(t_i)) = \frac{\exp(\beta^T x_i)}{\sum_{j \in R(t_i)} \exp(\beta^T x_j)} \tag{2.4}$$

La vraisemblance partielle est ensuite obtenue en multipliant ces contributions sur l'ensemble des événements observés :

$$L(\beta) = \prod_{i=1}^{m} \frac{\exp(\beta^T x_i)}{\sum_{j \in R(t_i)} \exp(\beta^T x_j)} \tag{2.5}$$

On maximise généralement le logarithme de cette expression pour obtenir l'estimateur des paramètres :

$$\hat{\beta} = \arg\max_{\beta} \log L(\beta) \tag{2.6}$$

Cette formulation présente deux avantages majeurs :

1. Elle permet d'intégrer naturellement les données censurées en ne tenant compte que de l'ordre des événements
2. Elle évite d'imposer une forme paramétrique sur le risque de base, ce qui confère au modèle une grande flexibilité tout en conservant une interprétation directe des coefficients via les **hazard ratios**.

Cette approche présente plusieurs avantages dans le contexte de la survie. Tout d'abord, elle permet de gérer naturellement les données censurées en se fondant uniquement sur l'ordre d'apparition des événements, sans nécessiter la valeur exacte du temps de survie pour chaque patient. Ensuite, le fait de ne pas spécifier de forme paramétrique pour le risque de base $\lambda_0(t)$ confère au modèle une flexibilité appréciable, tout en évitant les risques d'un mauvais ajustement à une dynamique temporelle qui pourrait varier selon les patients.

Interprétation via les Hazard Ratios Par ailleurs, les coefficients β conservent une interprétation clinique directe via les *hazard ratios*, définis comme :

$$HR_k = \exp(\beta_k) \tag{2.7}$$

ce qui permet d'estimer l'effet multiplicatif d'une variation d'une unité d'une covariable x_k sur le risque instantané de décès. Un $HR_k > 1$ traduit une augmentation du risque, tandis qu'un $HR_k < 1$ suggère un effet protecteur.
Ainsi, malgré l'émergence récente de méthodes d'apprentissage plus flexibles ou non paramétriques, le modèle de Cox demeure une référence en oncologie en raison de son compromis unique entre robustesse statistique, performance prédictive et interprétabilité clinique qui représente un critère essentiel lorsque les décisions médicales exigent transparence et justification.

V.2.3 Performance de référence

Ce modèle présente l'avantage d'être facilement interprétable, mais sa structure log-linéaire limite sa capacité à modéliser les interactions complexes entre gènes sans un feature engineering poussée. Le score de performance obtenu par ce modèle sur le jeu de test, calculé via la métrique IPCW C-index, est de :

$$\text{C-index}_{Cox} = 0.6541$$

Ce résultat de 0.6541 représente l'objectif minimal que mes modèles devront dépasser pour démontrer leur pertinence.

Initialement, j'ai exploré une modélisation classique via le modèle de Cox. Cette première itération a démontré une efficacité certaine, atteignant un score de 0.7524. Toutefois, face à la stagnation de ce score malgré mes optimisations, j'ai identifié la nécessité de changer de méthode. Le développement de l'architecture de Stacking et l'intégration du Boosting ont été la réponse directe à cette limitation, avec pour objectif explicite de dépasser ce seuil de performance.

Annexe C : Présentation des Algorithmes

Pour capturer toute la complexité des données biologiques (non-linéarités, interactions gène-gène) tout en garantissant la robustesse des prédictions face au bruit, j'ai misé sur la diversité. Mon architecture repose sur 9 modèles issus de 5 familles algorithmiques distinctes.

Cette hétérogénéité est cruciale pour le *Stacking* final : chaque algorithme voit les données sous un angle différent, permettant de corriger les biais individuels de chacun.

V.3.1 Histogram Gradient Boosting (HGBT)

V.3.1.1 Fondements Théoriques

Le Gradient Boosting est une méthode d'apprentissage d'ensemble qui construit un modèle prédictif fort par l'agrégation séquentielle de modèles faibles (généralement des arbres de décision).

Principe Mathématique : L'Approche Additive

Soit un jeu de données $\{(x_i, y_i)\}_{i=1}^{N}$. L'objectif est de trouver une fonction $\hat{F}(x)$ qui minimise l'espérance d'une fonction de perte $\mathcal{L}(y, F(x))$. Le modèle est construit de manière itérative. À l'étape m, on cherche à améliorer le modèle précédent $F_{m-1}(x)$ en ajoutant un nouvel estimateur $h_m(x)$:

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x) \tag{3.1}$$

où :

- ν est le taux d'apprentissage (*learning rate*), un hyperparamètre de régularisation ($0 < \nu \leq 1$).
- $h_m(x)$ est un arbre de régression entraîné pour prédire le **pseudo-résidu** (le gradient négatif de la fonction de perte) :

$$r_{i,m} = -\left[\frac{\partial \mathcal{L}(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{3.2}$$

L'Innovation "Histogram-Based"

L'algorithme standard du Gradient Boosting est coûteux en calcul car il nécessite de trier les valeurs de chaque variable continue pour trouver le meilleur point de coupure (complexité en $O(N \log N)$). L'approche *Histogram-Based* (inspirée de LightGBM) discrétise les variables continues en un nombre fixe de "bins" **avant** l'entraînement.

- Cela réduit la complexité de la recherche de seuil à $O(\text{bins})$, rendant l'algorithme extrêmement rapide sur de grands jeux de données.
- Cette discrétisation agit également comme une forme de régularisation implicite, réduisant le risque de sur-apprentissage sur le bruit des données continues (comme les valeurs exactes de VAF).

V.3.1.2 Mise en Œuvre dans le Projet

Dans mon architecture de Stacking, le HGBT est l'algorithme le plus sollicité (4 modèles sur 9) en raison de sa capacité à gérer nativement les valeurs manquantes et de sa flexibilité vis-à-vis des fonctions de perte.

1. **Modélisation du Risque Instantané (Classification)** Le modèle `gb1` est un *HistGradientBoostingClassifier* entraîné sur le jeu de données étendu temporellement.
   - **Objectif :** Estimer la probabilité conditionnelle $P(T < t + \Delta t | T > t)$.
   - **Fonction de Perte :** L'entropie croisée binaire (*Log-Loss*), adaptée à la classification binaire (Vivant/Décédé sur l'intervalle) :

$$\mathcal{L}_{log}(y, p) = -[y \log(p) + (1 - y) \log(1 - p)] \tag{3.3}$$

2. **Modélisation de la Durée de Survie (Régression)** Les modèles `gb2`, `gb3` et `gb4` sont des régresseurs appliqués directement à la cible corrigée $\log(1 + Y)$. J'ai diversifié les fonctions de perte pour capturer différentes propriétés de la distribution de survie :
   - **Perte de Poisson (`gb2`) :** Adaptée aux données de comptage ou aux événements rares. Elle pénalise fortement les erreurs sur les valeurs faibles, ce qui est pertinent pour identifier les patients à très haut risque (décès précoce).
   - **Perte Gamma (`gb3`) :** Conçue pour les variables continues positives avec une distribution asymétrique, ce qui correspond typiquement à la distribution des temps de survie en oncologie.
   - **Erreur Quadratique (MSE, `gb4`) :** La perte standard (L_2), qui se concentre sur la moyenne conditionnelle.

V.3.2 Perceptron Multi-Couches (MLP)

V.3.2.1 Fondements Théoriques

Le Perceptron Multi-Couches est un approximateur universel capable de modéliser des relations non-linéaires extrêmement complexes entre les variables d'entrée et la cible.

Architecture Neuronale

Le modèle est structuré en couches successives de neurones artificiels :

- Une **couche d'entrée** correspondant aux p variables explicatives (cliniques, génétiques, textuelles).
- Une ou plusieurs **couches cachées** qui opèrent des transformations non-linéaires.
- Une **couche de sortie** qui agrège les signaux pour produire la prédiction finale (temps de survie).

Principe Mathématique : La Propagation

Chaque neurone d'une couche l reçoit les sorties de la couche précédente $l-1$, calcule une combinaison linéaire pondérée, puis applique une fonction d'activation non-linéaire ϕ. Pour un vecteur d'entrée $a^{[l-1]}$, la sortie $a^{[l]}$ de la couche l est définie par :

$$a^{[l]} = \phi\left(W^{[l]}a^{[l-1]} + b^{[l]}\right) \tag{3.4}$$

où :

- $W^{[l]}$ est la matrice des poids synaptiques (à apprendre).
- $b^{[l]}$ est le vecteur de biais.
- ϕ est la fonction d'activation. J'utilise généralement la fonction **ReLU** (Rectified Linear Unit) définie par $\phi(z) = \max(0, z)$, qui permet de gérer efficacement la non-linéarité tout en facilitant l'apprentissage (pas de saturation du gradient).

Apprentissage : Rétropropagation du Gradient

L'entraînement consiste à minimiser une fonction de coût (ici l'erreur quadratique $\mathcal{L}_{MSE}$) en ajustant les poids W et b. Cela se fait par l'algorithme de la descente de gradient, où les erreurs sont propagées de la sortie vers l'entrée pour calculer les gradients partiels :

$$W^{[l]} \leftarrow W^{[l]} - \eta\frac{\partial\mathcal{L}}{\partial W^{[l]}} \tag{3.5}$$

où η est le taux d'apprentissage.

V.3.2.2 Mise en Œuvre et Apport dans le Projet

Dans mon architecture de Stacking, le MLP joue le rôle de spécialiste des interactions complexes, là où les modèles à base d'arbres (Boosting, RF) se concentrent sur des règles de décision hiérarchiques.

1. **Architecture Spécifique** J'ai configuré un réseau dense avec deux couches cachées de respectivement **100 et 50 neurones**. Cette structure en entonnoir force le modèle à compresser l'information et à extraire des caractéristiques latentes de haut niveau à partir des données brutes (VAF des gènes, scores cytogénétiques).

2. Importance du Pré-traitement (Scaling) Contrairement aux arbres de décision, les réseaux de neurones sont extrêmement sensibles à l'échelle des variables d'entrée. Une variable avec de grandes valeurs (ex: `WBC` ou `PLT`) pourrait écraser les autres dans le calcul de la combinaison linéaire. C'est pourquoi j'ai intégré le modèle dans un **Pipeline** incluant impérativement un `StandardScaler` en amont :

$$z = \frac{x - \mu}{\sigma}$$

Cette normalisation (moyenne 0, variance 1) est indispensable pour la convergence de l'algorithme d'optimisation.

3. Stratégie de Prédiction Le MLP est utilisé en mode régression sur la cible corrigée $\log(1 + Y)$. Comme la sortie du modèle est une estimation du temps de survie, j'inverse son signe ($-\hat{y}$) avant l'agrégation pour obtenir un **score de risque** (un temps de survie court correspond à un risque élevé, et inversement).

V.3.3 Régression par les K-Plus Proches Voisins (KNN)

V.3.3.1 Fondements Théoriques

L'algorithme des K-Plus Proches Voisins (*K-Nearest Neighbors*) est une méthode non-paramétrique basée sur l'analogie. L'hypothèse fondamentale est que des observations proches dans l'espace des caractéristiques auront des valeurs cibles similaires.

Principe Mathématique : La Topologie Locale

Soit un patient cible représenté par un vecteur x. L'algorithme identifie l'ensemble $\mathcal{N}_k(x)$ des k observations de l'ensemble d'entraînement les plus proches de x, selon une métrique de distance d (généralement la distance Euclidienne L_2) :

$$d(x, x_i) = \sqrt{\sum_{j=1}^{p} (x^{(j)} - x_i^{(j)})^2}$$

La prédiction $\hat{y}(x)$ est ensuite calculée comme une moyenne pondérée des cibles y_i des voisins :

$$\hat{y}(x) = \frac{\sum_{x_i \in \mathcal{N}_k(x)} w_i \cdot y_i}{\sum_{x_i \in \mathcal{N}_k(x)} w_i} \tag{3.6}$$

Dans ma configuration, j'utilise une pondération par l'inverse de la distance (`weights='distance'`), définie par $w_i = \frac{1}{d(x, x_i)}$. Cela permet de donner plus d'influence aux patients dont le profil biologique est quasi-identique à celui du patient cible, par rapport aux voisins plus éloignés.

V.3.3.2 Mise en Œuvre et Apport dans le Projet

Dans une architecture de Stacking dominée par les arbres de décision, le KNN apporte une diversité essentielle en basant ses prédictions sur la **similarité clinique** plutôt que sur des règles de coupure hiérarchiques.

1. Prétraitement Indispensable : Normalisation Non-Linéaire Le KNN est extrêmement sensible à l'échelle et à la distribution des variables. Une variable à forte amplitude (ex : taux de plaquettes, variant de 0 à 1000) dominerait totalement le calcul de la distance Euclidienne face à une variable génétique (VAF entre 0 et 1). Pour corriger cela, j'ai intégré dans le pipeline un **`QuantileTransformer`** avec une sortie normale.

- Cette transformation mappe la distribution de chaque variable vers une distribution Gaussienne standard ($\mathcal{N}(0, 1)$).
- Elle permet de rendre le calcul de distance robuste aux valeurs aberrantes et de mettre toutes les variables cliniques et génétiques sur un pied d'égalité géométrique.

2. Configuration du Voisinage J'ai fixé le nombre de voisins à $k = 40$. Ce choix, relativement élevé, permet de lisser les prédictions et de réduire la variance locale, évitant que la prédiction ne soit trop influencée par le destin atypique d'un seul patient isolé. Ce modèle agit ainsi comme un lisseur local dans mon ensemble final.

V.3.4 Forêts Aléatoires (Random Forest)

V.3.4.1 Fondements Théoriques

L'algorithme des Forêts Aléatoires (*Random Forest*) est une méthode d'ensemble qui repose sur le principe du **Bagging** (*Bootstrap Aggregating*). Contrairement au Boosting qui construit les arbres séquentiellement pour corriger les erreurs, le Bagging construit une multitude d'arbres **en parallèle** et de manière indépendante.

Principe Mathématique

Pour construire une forêt de B arbres, l'algorithme procède ainsi pour chaque arbre b :

1. **Bootstrap :** On tire un échantillon aléatoire D_b de même taille que le jeu de données original, avec remise (environ 63% des données uniques sont sélectionnées).
2. **Random Subspace :** Lors de la construction de l'arbre, à chaque nœud, on sélectionne aléatoirement un sous-ensemble de variables candidates pour trouver la meilleure coupure. Cela décorèle les arbres entre eux.

La prédiction finale $\hat{f}_{RF}(x)$ est la moyenne arithmétique des prédictions de tous les arbres $T_b(x)$:

$$\hat{f}_{RF}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{3.7}$$

Propriété clé : Si chaque arbre individuel a une variance élevée, la moyenne d'arbres décorrélés réduit considérablement la variance globale sans augmenter le biais. C'est ce qui rend le Random Forest particulièrement robuste au bruit.

La figure ci-dessous permet de vérifier la cohérence biologique de l'algorithme.

1. **Validation du Feature Engineering :** La variable `worst_gene` (issue de mon Target Encoding) arrive en tête du classement. Cela confirme que synthétiser le risque génétique global d'un patient est plus efficace que de traiter chaque mutation isolément.
2. **Facteurs Cliniques :** La présence marquée de biomarqueurs sanguins tels que le taux de plaquettes (`log_PLT`) et l'hémoglobine (`HB`) dans le top 5 valide la pertinence du modèle. Ces variables, reflets directs de l'état hématologique du patient, jouent un rôle pondérateur essentiel face aux mutations génétiques.
3. **Détail Moléculaire :** Le modèle identifie ensuite des mutations spécifiques (`vaf_...`) et des marqueurs cytogénétiques, prouvant qu'il combine à la fois le terrain du patient et l'agressivité biologique de la tumeur pour établir son pronostic.

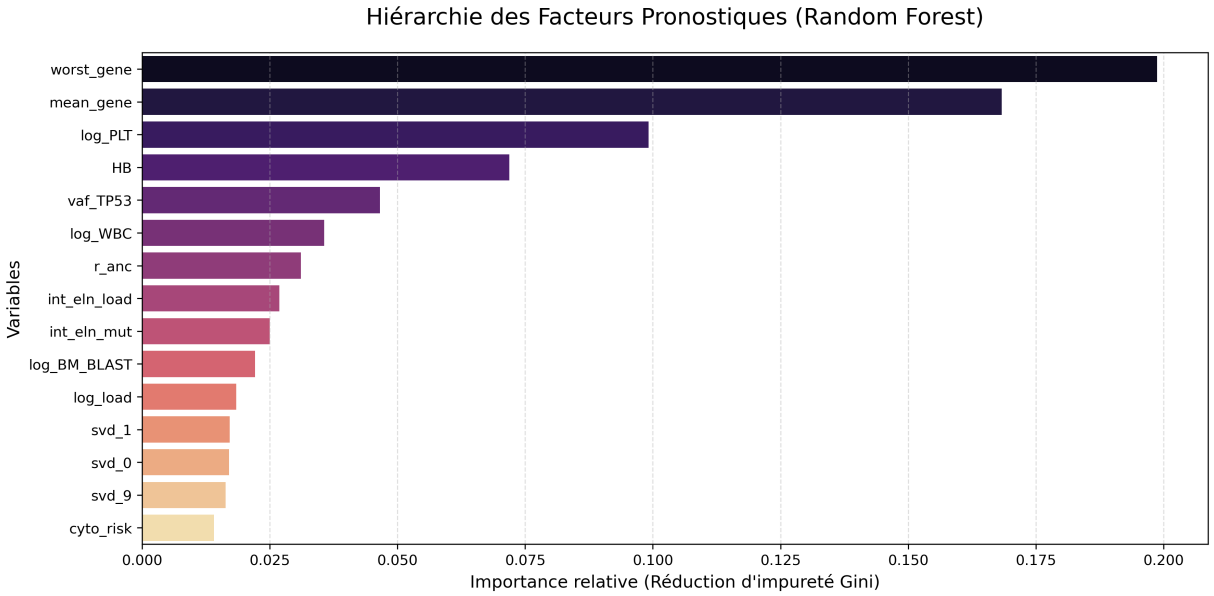

Figure V.3.1 – Importance relative des 15 variables les plus discriminantes (Critère de Gini)

V.3.4.2 Mise en Œuvre et Apport dans le Projet

Dans mon architecture, le Random Forest joue le rôle de **stabilisateur**. Là où le Gradient Boosting et le MLP cherchent la performance pure au risque de sur-apprendre, le Random Forest produit des prédictions plus conservatrices et fiables.

Diversification des Hyperparamètres

Pour enrichir le *Stacking*, j'ai entraîné deux variantes distinctes du modèle :

- **Modèle Standard (`rf`) :** Profondeur limitée à 9 et minimum de 4 échantillons par feuille. Il capture les tendances générales.
- **Modèle Profond (`rf_deep`) :** Profondeur poussée à 12 et minimum de 3 échantillons par feuille. Il est autorisé à capturer des interactions plus fines et spécifiques.

Ces modèles sont entraînés sur la cible de régression corrigée ($\log(1 + Y)$) obtenue via ma stratégie semi-supervisée.

V.3.5 Régression Lasso (Baseline Linéaire)

V.3.5.1 Fondements Théoriques

Le Lasso (*Least Absolute Shrinkage and Selection Operator*) est un modèle linéaire qui intègre une régularisation L_1. Il cherche à expliquer la cible y par une combinaison linéaire des variables x, tout en contraignant la somme des valeurs absolues des coefficients.

Principe Mathématique : La Parcimonie

L'algorithme cherche le vecteur de coefficients β qui minimise la fonction objectif suivante :

$$\min_{\beta} \left(\frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right) \tag{3.8}$$

où :

- Le premier terme est l'erreur quadratique classique (moindres carrés).
- Le second terme est la pénalité L_1, contrôlée par l'hyperparamètre α.

La géométrie de la contrainte L_1 a une propriété unique : elle force certains coefficients β_j à devenir **exactement nuls**. Le Lasso effectue donc une **sélection de variables** automatique, ne conservant que les prédicteurs les plus influents.

V.3.5.2 Mise en Œuvre et Apport dans le Projet

Bien que la relation entre la génétique et la survie soit complexe, certaines variables (comme l'âge, le taux de blastes ou le score ELN) ont un effet linéaire fort et direct.

Rôle de "Garde-Fou" Le Lasso sert de référence de base dans mon ensemble. Si les modèles complexes (MLP, Boosting) se perdent dans le bruit, le Lasso garantit que les signaux linéaires évidents sont correctement capturés.

Pré-traitement Robuste Les modèles linéaires sont sensibles aux valeurs aberrantes. Dans mon implémentation, j'ai précédé le Lasso d'un `RobustScaler`. Ce scaler centre et réduit les données en utilisant la médiane et l'écart interquartile (IQR) plutôt que la moyenne et l'écart-type, rendant le modèle insensibles aux valeurs extrêmes fréquentes dans les données cliniques (ex: taux de leucocytes explosif). J'ai utilisé un α faible (0.0001), permettant au modèle de conserver un large spectre de variables tout en filtrant le bruit non informatif.

Bibliographie

[1] Qube Research & Technologies (QRT), *Challenge Data : Prédiction de survie pour la Leucémie Myéloïde Aiguë*, Plateforme Challenge Data ENS, 2025.

[2] D. Micci-Barreca, *A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems*, ACM SIGKDD Explorations Newsletter, 2001.

[3] G. Salton, C. Buckley, *Term-weighting approaches in automatic text retrieval*, Information Processing Management, 1988.

[4] P. D. Allison, *Discrete-Time Methods for the Analysis of Event Histories*, Sociological Methodology, 1982.

[5] G. Ke et al., *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*, Advances in Neural Information Processing Systems (NIPS), 2017.

[6] Scikit-Learn User Guide/Scikit-learn Developers, *Histogram-Based Gradient Boosting*, Documentation officielle, Section 1.11.

[7] D. H. Wolpert, *Stacked Generalization*, Neural Networks, 1992.

[8] H. Döhner et al., *Diagnosis and management of AML in adults: 2017 ELN recommendations from an international expert panel*, Blood, 2017.