**Introduction :**

**Définition simple et directe** : Le cancer du poumon est une maladie caractérisée par une croissance incontrôlée de cellules anormales dans les poumons.

**Facteurs de risque** : Tabagisme, pollution de l'air, exposition à certains produits chimiques.

**Traitement :**

Si on dit que la **survie à 5 ans après chirurgie est de 50-60 %**, cela signifie que **sur 100 patients opérés, entre 50 et 60 seront encore en vie 5 ans après le traitement**.

**Les stades de cancer :**

 **Qu'est-ce qu'un stade de cancer ?**

* C’est une classification qui décrit la **taille de la tumeur et son extension** dans l’organisme.
* Le **staging** le plus utilisé est le **TNM** :
  + **T (Tumeur)** : Taille et localisation.
  + **N (Ganglions)** : Présence dans les ganglions lymphatiques.
  + **M (Métastases)** : Si le cancer s’est propagé à d’autres organes.
* On parle souvent de **stades 1 à 4** (1 = localisé, 4 = métastases).

 **Pourquoi c’est important ?**

* Plus le stade est élevé, plus le **pronostic est mauvais**.
* Un diagnostic précoce (stade 1 ou 2) améliore les chances de survie.

 **Transition vers l’IA et la prédiction**

* Si on pouvait prédire le risque de cancer grâce aux **données physiologiques et comportementales**, on pourrait **inciter les personnes à faire des examens plus tôt** et **détecter des cancers à un stade précoce**, ce qui améliorerait la prise en charge.

**Problématique : RAS**

**Methode :** RAS

**Présentation des doonées** RAS

**Importation des données :** RAS

**Présentation des données :**

**Échelles Similaires des Variables**

Dans notre jeu de données, les variables sont toutes discrètes, avec des valeurs allant de 0 à 7 pour certaines et de 0 à 9 pour d'autres. Ce que l'on appelle des **échelles similaires** signifie que les plages de valeurs de ces variables ne diffèrent pas de manière excessive. Par exemple, une variable qui va de 0 à 7 est sur une échelle proche d'une autre qui va de 0 à 9. Cette similarité dans les plages de valeurs est importante car elle permet aux modèles de machine learning de traiter toutes les variables de manière équitable, sans que certaines variables n'aient plus d'impact simplement à cause de leur plage de valeurs plus large.

De plus, comme toutes nos variables ont un écart type similaire, cela signifie qu'elles présentent une **dispersion similaire autour de leur moyenne**. En d'autres termes, elles varient de façon comparable, ce qui facilite leur interprétation par les modèles. Si les variables avaient des écarts types très différents, cela aurait pu entraîner des déséquilibres dans l'importance donnée à chacune d'elles pendant l'entraînement des modèles.

Ce type d'échelle similaire entre les variables simplifie la gestion des données par les modèles, notamment les modèles comme les arbres de décision, les KNN, ou même les SVM. Ces derniers sont capables de mieux capturer les relations entre les variables sans être biaisés par l'échelle des valeurs, ce qui explique en partie les **performances exceptionnelles** que nous avons observées dans nos résultats.

**Les boxplots :**

**1. Position des médianes**

* Compare la **médiane** (la ligne au centre de chaque boîte) pour chaque classe (High, Medium, Low). Si les médianes des différentes classes sont nettement séparées, cela suggère que les variables ont un pouvoir discriminant et que le modèle pourrait facilement distinguer les différentes classes.
* Si les médianes se chevauchent fortement, cela pourrait indiquer que certaines variables n’apportent pas de distinction nette entre les classes, ce qui peut rendre la tâche de prédiction plus difficile.

**2. Ampleur des boîtes (Interquartile Range - IQR)**

* **L’étendue de chaque boîte** te montre l'**écart interquartile** (IQR), c'est-à-dire la dispersion des 50% des données autour de la médiane. Si les boîtes sont très larges pour une classe, cela signifie qu'il y a une grande variabilité au sein de cette classe, ce qui pourrait indiquer une **hétérogénéité** ou une **incertitude** dans les données pour cette classe.
* Si les boîtes sont petites, cela pourrait signifier que les valeurs sont bien regroupées autour de la médiane, ce qui pourrait être plus facile à prédire.

**3. Présence de valeurs aberrantes (Outliers)**

* Regarde si certaines variables contiennent des **valeurs aberrantes** (les points situés en dehors des moustaches). Ces points sont des **valeurs extrêmes** qui peuvent influencer les résultats des modèles de manière significative. Il peut être utile de les analyser plus en profondeur.
* Une forte présence d'outliers dans une ou plusieurs classes pourrait rendre l'apprentissage plus complexe, surtout pour des modèles comme **KNN** ou **SVM**, qui peuvent être sensibles aux valeurs extrêmes.

**4. Chevauchement des boîtes entre les classes**

* Si les boîtes des différentes classes se **chevauchent beaucoup**, cela peut indiquer que les variables ne sont pas très discriminantes et que le modèle pourrait avoir des difficultés à séparer les classes.
* Si les boîtes sont **bien séparées**, cela signifie que les variables sont potentiellement très utiles pour distinguer les classes.

**5. Symétrie ou asymétrie des distributions**

* Si les boîtes sont **asimétriques**, cela pourrait indiquer une **distribution biaisée** des données pour cette variable, ce qui pourrait affecter les performances des modèles.
* Une asymétrie marquée pourrait également suggérer qu’une **transformation** (comme la log ou racine carrée) pourrait être utile pour stabiliser la variance ou rendre la distribution plus normale.

**6. Comparaison entre les classes**

* Compare la forme générale des boxplots pour les différentes classes. Par exemple, pour une variable, si les boxplots pour **High** et **Medium** sont très similaires et que celui de **Low** est très différent, cela pourrait indiquer qu'il est plus facile de distinguer **Low** des autres classes que de différencier **High** et **Medium**.

L'asymétrie d'une distribution fait référence à la façon dont les données sont réparties autour de la moyenne. Si une distribution est **asimétrique**, cela signifie qu'il y a une **queue** plus longue d'un côté de la moyenne que de l'autre.

**Types d'asymétrie :**

* **Asymétrie positive (ou à droite)** : La queue de la distribution est plus longue du côté des valeurs plus élevées. Cela signifie qu'il y a **plus de valeurs faibles** et seulement quelques valeurs extrêmement élevées. Une distribution comme celle-ci est souvent appelée **distribution étalée à droite**.
* **Asymétrie négative (ou à gauche)** : La queue de la distribution est plus longue du côté des valeurs plus faibles. Cela signifie qu'il y a **plus de valeurs élevées** et quelques valeurs extrêmement faibles. Ce type de distribution est appelé **distribution étalée à gauche**.

**Pourquoi l’asymétrie peut être importante pour tes modèles :**

1. **Modèles sensibles à la normalité** : Certains modèles, comme la régression linéaire, les modèles de classification linéaire (par exemple LDA ou QDA) et **Naive Bayes**, font l'hypothèse que les données suivent une **distribution normale**. Si tes variables ont une forte **asymétrie**, cela peut entraîner une **baisse de performance** de ces modèles, car la relation entre les variables ne sera pas linéaire et ne correspondra pas bien à l’hypothèse de normalité.
2. **Stabilisation de la variance** : Quand une variable est **fortement asymétrique**, sa variance peut être très élevée, ce qui peut poser des problèmes pour les modèles qui sont sensibles à la variance. Une **transformation** de la variable peut aider à réduire cette asymétrie et stabiliser la variance.

**Transformation pour corriger l’asymétrie :**

* **Logarithme (log)** : Si tu as une **asymétrie positive**, où les valeurs sont étendues à droite, une transformation logarithmique peut aider. Elle compresse les grandes valeurs, ce qui réduit l'impact des **outliers** et rend la distribution plus proche de la normale. Par exemple, une variable mesurant les revenus pourrait être log-transformée, car les revenus très élevés peuvent fausser la distribution.
* **Racine carrée** : Une transformation de la racine carrée peut également être utile pour les variables ayant une asymétrie positive, mais elle est généralement moins forte qu'un logarithme. Elle est utilisée quand les valeurs extrêmes ne sont pas aussi sévères mais doivent être réduites.
* **Transformation inverse (1/x)** : Pour les variables avec une asymétrie négative, une transformation inverse peut être utile. Elle va inverser la distribution, réduisant ainsi l'impact des faibles valeurs.

**Analyse des Barplots**

En complément des boxplots, nous avons utilisé des **barplots** pour examiner la répartition des variables par classe cible (High, Medium, Low). En coloriant les barres en fonction de la proportion de chaque classe, plusieurs observations importantes ont été faites :

1. **Répartition des variables par classe** :  
   Les barplots ont montré une variation nette dans la proportion des classes (High, Medium, Low) pour certaines variables. Cela a permis de mieux visualiser comment les valeurs d’une variable sont associées à chaque niveau de risque. Par exemple, certaines variables montrent des proportions **très déséquilibrées** entre les classes, ce qui peut être un indicateur de leur forte **discrimination** entre les niveaux de risque.
2. **Discrimination des classes** :  
   Certaines variables ont une répartition qui permet de **distinguer clairement** les classes entre elles. Lorsque la couleur des barres diffère fortement entre les classes, cela suggère que la variable est un bon indicateur de la classe cible.
3. **Identification des variables redondantes** :  
   Pour certaines variables, la répartition des classes est **très similaire** sur les barplots. Cela pourrait indiquer que ces variables apportent peu de valeur discriminante et pourraient être **moins utiles** pour la prédiction.

**Corrplot :**

**Corrélations fortes** entre de nombreuses variables : Un grand nombre de variables sont **fortement corrélées** entre elles, ce qui pourrait poser des problèmes de **multicolinéarité**. Cela signifie que certaines variables mesurent des phénomènes très similaires, ce qui rend les modèles sensibles à **l’instabilité des paramètres**. Cette instabilité pourrait nuire à la capacité des modèles à **généraliser** et à faire de bonnes prédictions sur des données non vues.

* **Problèmes de surajustement** : La forte corrélation entre les variables peut également entraîner un phénomène de **surajustement** (overfitting). Le modèle pourrait apprendre des **relations redondantes**, qui n’apportent pas d’information nouvelle, mais augmentent la complexité du modèle, le rendant moins robuste et moins capable de s'adapter à de nouvelles données.

**Equilibre des classes :**

En examinant la répartition des classes (High, Medium, Low), nous pouvons constater que les classes sont relativement **équilibrées** :

* **High** : 365 individus
* **Low** : 303 individus
* **Medium** : 332 individus

Cela signifie que chaque classe contient un nombre assez similaire d'individus, ce qui est un bon point pour l'entraînement des modèles de classification. En effet, un **déséquilibre** important entre les classes aurait pu biaiser le modèle, en faisant en sorte que le modèle privilégie la classe majoritaire.

**Implications pour les modèles :**

L'équilibre des classes permet de **réduire les risques de biais** dans les prédictions et assure que chaque classe est suffisamment représentée pour être bien apprise par les modèles. Cela nous aide à construire des modèles plus **justes** et **robustes**.

**ACP :**

**Tab inertie :**

**Analyse en Composantes Principales (ACP)**

Pour réduire la dimensionnalité et mieux comprendre la structure des données, nous avons réalisé une **Analyse en Composantes Principales (ACP)**.

* **Axe 1 et Axe 2** : Ces deux axes capturent ensemble **53,4%** de la variance totale des données, ce qui signifie que **plus de la moitié de l'information** présente dans les données peut être résumée par ces deux composantes.
  + Cela indique que ces deux axes sont **suffisamment représentatifs** des données et peuvent être utilisés pour mieux comprendre la structure sous-jacente des variables.

**Approche choisie :**

Nous avons décidé de **focaliser notre analyse** sur ces deux premières dimensions, qui semblent être les plus significatives pour la classification et la réduction de la complexité des données.

**Graph VAR :**

**Cercle des Variables (ACP)**

En analysant le cercle des variables issu de l'ACP, on peut observer quelques points intéressants :

* **Tendance générale à droite** : La majorité des flèches des variables pointent vers la droite, ce qui pourrait indiquer que les **individus présentant un risque plus élevé de cancer** semblent avoir des caractéristiques similaires associées à ces variables. En d'autres termes, plus une variable est proche de la droite, plus elle pourrait être associée à un risque **plus élevé** de développer un cancer.
* **Corrélation entre variables** : Les **flèches très proches** les unes des autres indiquent qu'il existe une **forte corrélation** entre ces variables. Cela signifie que ces variables mesurent des informations similaires et peuvent être redondantes dans la prédiction du risque de cancer.

**Implications :**

Cette observation nous aide à identifier quelles variables sont **les plus discriminantes** pour le modèle et à comprendre les **relations** entre elles.

**Graph ind :** RAS

**Modèles :**

**Métriques de Performance**

**Étant donné le contexte médical et l’importance de détecter les cas à haut risque de cancer (High), j’ai décidé de privilégier certaines métriques pour évaluer la performance des modèles de classification. Les deux principales métriques que j’ai utilisées sont :**

1. **Recall (Rappel) :**
   * **Le recall mesure la capacité du modèle à détecter correctement les cas positifs (c'est-à-dire les individus à haut risque).**
   * **Dans un contexte médical, privilégier le recall est crucial, car il est plus important de détecter le plus grand nombre de cas de cancer possible, même si cela signifie une légère augmentation des faux positifs (patients faussement identifiés comme malades). Cela permet de s'assurer qu'on ne rate pas de cas importants.**
2. **Précision :**
   * **La précision mesure la proportion de vrais positifs parmi tous les cas identifiés comme positifs par le modèle.**
   * **Dans ce cas, la précision est moins prioritaire que le recall, car il est plus acceptable dans le cadre médical de classer un patient à tort comme malade (faux positif) que de manquer un cas de cancer (faux négatif).**
3. **F1-Score :**
   * **Le F1-score est la moyenne harmonique entre la précision et le recall. Il permet de trouver un équilibre entre ces deux métriques, en évitant de privilégier uniquement l’une au détriment de l’autre.**
   * **C’est une bonne mesure de performance lorsqu’il y a un compromis entre la précision et le recall.**
   * **L'équation du F1-score :**
4. **Courbe ROC et AUC (Area Under the Curve) :**
   * **La courbe ROC est un graphique qui montre la relation entre le taux de vrais positifs (TPR, ou recall) et le taux de faux positifs (FPR). L’AUC est l'aire sous la courbe ROC, et plus elle est proche de 1, mieux le modèle est.**
   * **Un AUC élevé indique que le modèle fait une bonne distinction entre les classes, ce qui est particulièrement important dans un contexte de classification de risques.**
5. **Matrice de Confusion :**
   * **La matrice de confusion est un tableau qui permet de visualiser les résultats d’un modèle de classification. Elle présente les vrais positifs, faux positifs, vrais négatifs, et faux négatifs.**

**Cas de Multiclass :**

**Dans une tâche de classification multiclass, chaque classe (High, Medium, Low) aura ses propres vrais positifs, faux positifs, et faux négatifs. Voici les définitions :**

* **Vrais Positifs (VP) : Nombre d’individus correctement classés dans la classe prédite.**
* **Faux Positifs (FP) : Nombre d’individus incorrectement classés dans une classe, alors qu'ils appartiennent à une autre classe.**
* **Faux Négatifs (FN) : Nombre d’individus qui sont réellement dans une classe, mais qui ont été classés dans une autre.**
* **Vrais Négatifs (VN) : Nombre d’individus qui n’appartiennent pas à la classe, et qui sont correctement classés en dehors de cette classe.**

**Prétraitement des Données**

Pour corriger les problèmes identifiés précédemment (fortes corrélations et présence d'outliers), j'ai appliqué plusieurs étapes de prétraitement avec la fonction **recipe** de la bibliothèque **recipes** en R. Voici les étapes que j'ai utilisées pour préparer les données avant l'entraînement des modèles :

1. **Suppression de la multicolinéarité** :
   * J'ai utilisé **step\_corr** pour supprimer les variables hautement corrélées entre elles. En effet, la présence de variables fortement corrélées peut entraîner des problèmes de **multicolinéarité**, rendant les modèles instables et affectant leur capacité à généraliser correctement.
   * J’ai fixé un **seuil dynamique** (tune()) pour déterminer quelles variables doivent être supprimées en fonction de leur degré de corrélation.
2. **Gestion des outliers avec Mahalanobis** :
   * J’ai appliqué **step\_outliers\_maha**, qui permet d'identifier les outliers à l’aide de la distance de **Mahalanobis**. Cette méthode est particulièrement utile pour détecter les points de données qui sont **éloignés** de la distribution générale de manière multivariée.
   * L'idée ici est de détecter les individus qui, par leurs valeurs extrêmes, peuvent influencer de manière disproportionnée l’entraînement des modèles.
3. **Identification et gestion des outliers supplémentaires** :
   * J’ai ajouté une étape avec **step\_outliers\_lookout**, qui permet d'identifier d'autres outliers potentiels. Cette étape est appliquée sur toutes les variables numériques, en excluant celles qui ont été déjà traitées comme des **outliers** (grâce à contains(r"(.outliers)")).
4. **Suppression des outliers** :
   * Une fois les outliers identifiés, j'ai utilisé **step\_outliers\_remove** pour **supprimer** ces valeurs extrêmes. La suppression est effectuée selon un **score de dropout** (score\_dropout = tune()), et l’**agrégation** des valeurs restantes est effectuée par **moyenne** (fonction aggregation\_function = "mean").
   * Cette étape permet d’avoir un modèle plus robuste et de garantir qu’il ne soit pas perturbé par des valeurs aberrantes qui pourraient nuire à sa capacité à généraliser.

**Modalité d’Études de la Courbe ROC**

Pour évaluer les performances des modèles de classification dans un contexte **multiclass** (avec les classes **High**, **Medium**, et **Low**), j’ai utilisé les courbes ROC et l'AUC (Area Under the Curve). Voici les étapes et les concepts clés que j'ai appliqués :

**1. Courbes ROC One-vs-All**

* En classification multiclass, chaque classe est traitée indépendamment par rapport aux autres. Cela s'appelle le **One-vs-All** (ou **One-vs-Rest**).
* Pour chaque classe (par exemple, High), je considère cette classe comme **positive** et toutes les autres classes comme **négatives**.
* Une courbe ROC est ensuite tracée pour chaque classe, représentant la **relation entre le taux de vrais positifs** (True Positive Rate - TPR) et le **taux de faux positifs** (False Positive Rate - FPR).
* L'objectif est de mesurer, pour chaque classe, la capacité du modèle à bien identifier les individus appartenant à cette classe, tout en minimisant le nombre de faux positifs (autres classes mal classées comme appartenant à cette classe).

**2. AUC de la Courbe ROC (One-vs-All)**

* **AUC** (Area Under the Curve) mesure la capacité du modèle à **séparer les classes**. Une **AUC plus élevée** indique que le modèle a une meilleure capacité à distinguer les classes.
* Pour chaque courbe ROC (One-vs-All), j’ai calculé l’AUC, ce qui donne une idée précise de la performance du modèle pour chaque classe individuelle.

**3. AUC Macro (Moyenne des Courbes ROC)**

* L'**AUC macro** est la moyenne des AUC de toutes les classes. Cela permet d'obtenir une mesure globale de la performance du modèle dans un contexte multiclass.
* Cette approche est utile car elle **pénalise moins les classes minoritaires** et permet de comparer les modèles de manière plus équitable entre toutes les classes.

**4. Interprétation des Résultats**

* En analysant les courbes ROC et leurs AUC respectives, je peux évaluer quel modèle **distinguait le mieux** les différentes classes.
* Une **AUC proche de 1** indique que le modèle performe bien, tandis qu'une **AUC proche de 0.5** suggère que le modèle ne fait guère mieux qu'un modèle aléatoire.

**QDA - Quadratic Discriminant Analysis**

Le modèle **QDA** a été l'un des premiers que j'ai utilisés dans ce travail, car il est bien adapté à des données où les classes sont supposées avoir des distributions normales mais avec des **variances différentes**. Cependant, lors de l'entraînement du modèle, j'ai rencontré une **erreur spécifique** :

**"groupe High n'est pas de rang plein"**.

**1. Problème de Rang Non Plein**

* Cette erreur survient lorsque le modèle QDA essaie de calculer la matrice de covariance pour une classe (dans ce cas, la classe **High**), mais que les données pour cette classe ne couvrent pas suffisamment de variabilité. Autrement dit, il y a des **variables redondantes** ou certaines variables qui ne varient pas suffisamment, ce qui rend la matrice de covariance **singulière** et donc **non inversible**.
* Le problème peut également apparaître si le nombre de points d’observation dans cette classe est trop faible par rapport au nombre de variables, ce qui empêche le modèle d’apprendre correctement la relation entre les variables.

**2. Solution Tentée : Tuning des Paramètres**

* **Pour résoudre ce problème**, j'ai tenté de **tuner certains hyperparamètres**, notamment en ajustant le **seuil** de décision et en introduisant un **dropout** afin d'augmenter la robustesse du modèle face à des données peu variées ou fortement corrélées.
* Ces ajustements visent à rendre le modèle plus flexible et à éviter l'overfitting (ou ajustement excessif) sur des classes avec des caractéristiques trop similaires ou insuffisantes.

**3. Stratégie de Validation des Modèles**

* Au départ, j'ai **priorisé l'accuracy** comme critère d’évaluation pour le modèle, car cela permet de voir dans quelle mesure le modèle est capable de classer correctement les données dans l'ensemble, sans se soucier spécifiquement des classes minoritaires.
* **Ensuite, pour équilibrer les performances** et mieux traiter les classes déséquilibrées (par exemple, si la classe High est sous-représentée), j'ai mis l'accent sur le **recall** (rappel). Cela permet de s’assurer que le modèle est capable de détecter un maximum de cas positifs pour chaque classe, en particulier pour les classes moins fréquentes.

Les modèles

**1. LDA (Linear Discriminant Analysis)**

* **Principe** : LDA cherche à maximiser la séparation entre les classes en trouvant une ou plusieurs **fonctions linéaires** qui permettent de prédire la classe d'un individu.
* **Utilisation** : Idéal lorsque les données suivent une distribution normale et que les classes ont une variance similaire.
* **Avantages** : Simple, rapide, et efficace avec peu de données.
* **Limite** : Peut être sensible à des outliers et à des relations non linéaires.

**2. QDA (Quadratic Discriminant Analysis)**

* **Principe** : Comme LDA, mais avec la différence que QDA permet une **variance différente** pour chaque classe. Cela signifie qu’il peut modéliser des frontières de décision **quadratiques** entre les classes.
* **Utilisation** : Adapté lorsque les classes ont des variances différentes.
* **Avantages** : Meilleure flexibilité que LDA pour les données avec des variances hétérogènes.
* **Limite** : Plus sensible aux petites tailles d’échantillons et aux corrélations élevées entre les variables.

**3. KNN (K-Nearest Neighbors)**

* **Principe** : Le modèle KNN classe un individu en fonction de la majorité des **k voisins les plus proches** dans l’espace des caractéristiques. La distance entre les individus est généralement calculée à l’aide de la distance euclidienne.
* **Utilisation** : Simple et efficace, surtout pour des jeux de données avec des frontières de décision complexes.
* **Avantages** : Facile à comprendre et à implémenter, pas de phase d’entraînement.
* **Limite** : Lenteur pour des jeux de données très grands, sensible au bruit et aux caractéristiques inutiles.

**4. Naive Bayes**

* **Principe** : Le modèle Naive Bayes repose sur le théorème de **Bayes**, avec l'hypothèse que les **variables sont indépendantes** les unes des autres. Il calcule la probabilité de chaque classe donnée les caractéristiques d'un individu.
* **Utilisation** : Très efficace lorsque les variables sont conditionnellement indépendantes.
* **Avantages** : Rapide, efficace, et peu coûteux en calculs, surtout avec de grandes quantités de données.
* **Limite** : L'hypothèse d'indépendance est rarement vraie, ce qui peut réduire la précision.

**5. SVM (Support Vector Machine)**

* **Principe** : SVM cherche à trouver une **frontière de séparation** (ou hyperplan) qui sépare les classes avec la **plus grande marge** possible. Il peut utiliser un noyau pour permettre des séparations non linéaires.
* **Utilisation** : Bien adapté pour les jeux de données complexes et de grande dimension.
* **Avantages** : Très puissant pour les données de dimension élevée, très bonne performance avec des marges larges.
* **Limite** : L'optimisation peut être longue et coûteuse, surtout pour de grandes quantités de données.

**6. SVM Linéaire**

* **Principe** : C’est une version simplifiée de SVM qui utilise un **hyperplan linéaire** pour séparer les classes.
* **Utilisation** : Idéal quand les données sont linéairement séparables.
* **Avantages** : Rapide à entraîner, facile à interpréter.
* **Limite** : Moins performant lorsque les classes sont complexes et non linéaires.

**7. SVM Radial (RBF)**

* **Principe** : Utilise un noyau **radial** (fonction de base radiale) pour transformer les données dans un espace à plus haute dimension, permettant de séparer des classes non linéaires.
* **Utilisation** : Utilisé quand les données sont non linéaires et complexes.
* **Avantages** : Très puissant pour des données non linéaires et complexes.
* **Limite** : Sensible aux paramètres du noyau, et peut être plus coûteux en calculs.

**8. Decision Tree (Arbre de Décision)**

* **Principe** : Un arbre de décision divise les données en fonction de caractéristiques qui offrent la meilleure séparation entre les classes, en créant des **noeuds** et en suivant des règles de décision.
* **Utilisation** : Idéal pour une interprétation facile des décisions du modèle.
* **Avantages** : Simple à comprendre, rapide à entraîner, peut gérer des données numériques et catégoriques.
* **Limite** : Sensible à l'overfitting (surchargé de détails), sensible aux petites variations dans les données.

**9. Random Forest**

* **Principe** : Ensemble d’arbres de décision, où chaque arbre est construit avec une partie aléatoire des données et des caractéristiques. La prédiction finale est obtenue par **vote majoritaire** des arbres.
* **Utilisation** : Pratique pour éviter l’overfitting des arbres de décision individuels.
* **Avantages** : Moins sensible à l’overfitting, robuste, offre des bonnes performances.
* **Limite** : Moins interprétable que les arbres de décision simples, peut être coûteux en calculs.

**10. Boosting (par exemple, Gradient Boosting)**

* **Principe** : Le boosting consiste à entraîner plusieurs modèles faibles (par exemple, des arbres de décision simples) et à les combiner en un modèle plus puissant en ajustant les erreurs des modèles précédents de manière séquentielle.
* **Utilisation** : Efficace pour augmenter les performances des modèles faibles.
* **Avantages** : Très puissant, particulièrement pour des données complexes, souvent meilleur que les modèles individuels.
* **Limite** : Peut être sujet à l’overfitting si trop d’arbres sont ajoutés, nécessite un réglage fin des paramètres.

**Métriques de Performance**

**1. Accuracy (Précision globale)**

* **Définition** : C'est la proportion des prédictions correctes faites par le modèle, peu importe la classe.
* **Interprétation** : Une **accuracy de 86.1%** signifie que 86,1% des prédictions faites par le modèle sont correctes.

**2. F1-Score (moyenne pondérée)**

* **Définition** : Une mesure qui combine à la fois la **précision** et le **rappel**. Il est particulièrement utile quand les classes sont déséquilibrées, en équilibrant les faux positifs et faux négatifs.
* **Interprétation** : Un **F1-Score de 86.3%** montre que le modèle équilibre bien la précision et le rappel, et obtient des performances solides.

**3. Recall (Rappel)**

* **Définition** : C'est la capacité du modèle à identifier correctement les vrais positifs (les éléments pertinents) parmi tous les éléments pertinents disponibles.
* **Interprétation** : Un **recall de 86.1%** signifie que le modèle réussit à identifier correctement 86.1% des cas positifs dans chaque classe.

**4. Precision (Précision)**

* **Définition** : Il s'agit de la proportion des prédictions positives qui sont réellement correctes. Cela mesure la "pureté" des prédictions positives.
* **Interprétation** : Un **précision de 87.9%** indique que parmi toutes les prédictions faites par le modèle pour chaque classe, 87.9% sont effectivement correctes.

**5. Specificity (Spécificité)**

* **Définition** : La capacité du modèle à identifier correctement les négatifs (les éléments non pertinents).
* **Interprétation** : Un **spécificité de 93.7%** montre que le modèle réussit à bien identifier les éléments qui n'appartiennent pas à chaque classe.

**6. ROC-AUC (Area Under the Curve)**

* **Définition** : Mesure la capacité du modèle à distinguer entre les classes. L'**AUC** est la surface sous la courbe ROC (Receiver Operating Characteristic), et une **AUC de 92.0%** signifie que le modèle a une excellente capacité à différencier les classes.

**7. Error (Erreur)**

* **Définition** : C'est la proportion d'erreurs faites par le modèle. C'est l'inverse de l'accuracy.
* **Interprétation** : Un **error de 13.9%** signifie que 13.9% des prédictions du modèle sont incorrectes.

Pk les courbes roc sont angulaires :

**1. Classification multiclass (One-vs-All)**

* Dans un problème de classification multiclass, la méthode **One-vs-All** est souvent utilisée, où chaque classe est comparée aux autres combinées. Cela crée plusieurs courbes ROC, une pour chaque classe.
* Les courbes ROC dans ce cadre peuvent devenir angulaires ou en escalier car, pour certaines classes, les prédictions peuvent être très "décisives" (prédictions très proches de 0 ou 1), ce qui crée des sauts brutaux dans la courbe au lieu d'une transition plus lisse. Ce phénomène est particulièrement visible lorsque le modèle est extrêmement confiant sur certaines classes, mais avec moins de variance dans les scores de probabilité.

**2. Scores de probabilité peu variés**

* Si tes modèles de classification génèrent des **probabilités de classe très tranchées** (presque toutes les prédictions sont proches de 0 ou 1), cela peut provoquer des transitions nettes dans la courbe ROC. Les courbes devraient idéalement être lisses si les probabilités de sortie du modèle étaient plus variées.
* Un modèle qui est trop confiant dans ses prédictions, même si elles sont incorrectes, peut donner une courbe ROC qui a une forme angulaire ou en escalier.

**3. Présence de nombreux "sauts" dans la probabilité prédite**

* Si la sortie de probabilité de ton modèle (pour chaque classe) est générée de manière un peu trop « binaire » (soit très proche de 0, soit très proche de 1), chaque changement de seuil sur la courbe ROC provoque un grand saut, donnant un effet en escalier.
* Cela peut aussi être lié à un mauvais étalonnage des probabilités (le modèle n'est pas suffisamment "lisse" dans ses prédictions de probabilité). Les valeurs de sortie du modèle sont trop éloignées des valeurs de seuils intermédiaires, entraînant ces angles.

**4. Nombre de classes ou imbalance**

* Si le nombre de classes dans ton problème est élevé ou si la distribution des classes est déséquilibrée, la courbe ROC peut avoir une forme angulaire. Par exemple, les classes avec un nombre élevé de prédictions positives peuvent avoir une courbe plus « lisse », tandis que d'autres classes avec moins d'exemples peuvent montrer des transitions plus nettes, créant un effet de "pente" ou d'escalier.

**5. Seuils de prédiction choisis**

* Le fait que les courbes soient angulaires peut aussi être un effet des **seuils de classification** choisis pour calculer les valeurs de vraisemblance (probabilité). Si tu utilises des seuils fixes pour décider quelle classe prédire, cela peut rendre la courbe plus abrupte.
* Par exemple, si tu choisis de classer un échantillon comme appartenant à une classe si sa probabilité est supérieure à 0.8, mais moins de 0.8 sinon, tu peux avoir un changement brusque de la courbe ROC.

**6. Problème d'overfitting ou de modèle trop simple**

* Si ton modèle est en sur-apprentissage (overfitting), il peut devenir trop spécifique aux données d'entraînement et ne pas généraliser correctement aux nouvelles données. Cela peut provoquer des prédictions très sûres mais incorrectes, ce qui peut créer des changements brusques dans la courbe ROC.
* D'un autre côté, un modèle trop simple pourrait ne pas être assez différencié entre les classes et donner des courbes avec des changements moins graduels.