Centrale
CASABLANCA
Coding Week
RAPPORT

Rapport de projet3 Groupe15

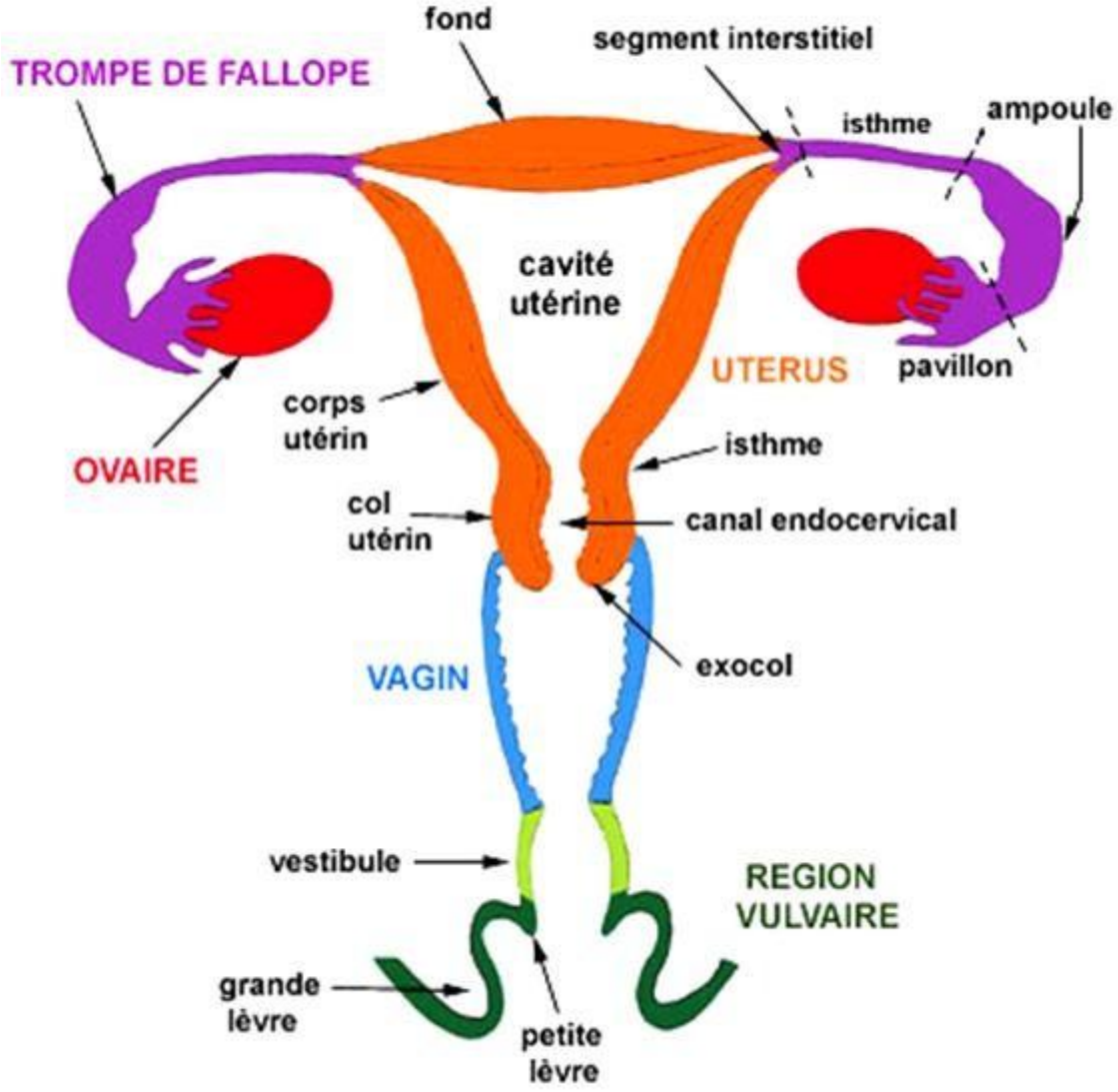
Risque du Cancer du col de l'utérus

Tuteurs : Kawtar Zerhouni
et Hermann Agossou

Groupe 15 :

- **GUINKO Fognon Anais Aurelie Sarah**
- **OUATTARA Mory Désiré**
- **OUEDRAOGO Mahamoudou**
- **RABO Gueriatou**
- **TIENDREBEOGO Baowendsom Arlette**

Année Scolaire
2024-2025



Visualisation de l'utérus

I.Introduction

- ***L'intelligence artificielle (IA) révolutionne de nombreux secteurs, et la santé ne fait pas exception. En intégrant des technologies avancées d'IA, le domaine médical connaît des transformations significatives qui améliorent la qualité des soins, optimisent les diagnostics et personnalisent les traitements. L'IA permet***

d'analyser de vastes quantités de données médicales, d'identifier des schémas complexes et de fournir des prédictions précises, ce qui aide les professionnels de la santé à prendre des décisions éclairées.

- *L'une des applications les plus prometteuses de l'IA en santé est le développement d'outils de support à la décision clinique. Ces outils utilisent des algorithmes de machine Learning pour évaluer les risques de maladies, recommander des traitements et même prédire les résultats des interventions médicales. Par exemple, des modèles explicables comme SHAP (Shapley Additive explantations) permettent non seulement de faire des prédictions, mais aussi d'expliquer les facteurs qui influencent ces prédictions, offrant ainsi une transparence essentielle pour les médecins et les patients.*
- *En outre, l'IA contribue à la gestion des dossiers médicaux électroniques, à la surveillance des patients en temps réel et à la recherche médicale. Des plateformes comme "Mon espace santé" en France permettent aux patients de gérer leurs informations de santé de manière sécurisée et de les partager avec leurs professionnels de santé*
- *Ces avancées montrent comment l'IA peut améliorer l'efficacité des soins de santé tout en assurant une meilleure coordination entre les différents acteurs du système de santé.*
- *Dans ce contexte, notre projet vise à développer une application de support à la décision médicale pour l'évaluation du risque de cancer du col de l'utérus. En utilisant des techniques de machine Learning explicables, nous cherchons à fournir aux médecins des outils précis et transparents pour améliorer la qualité des diagnostics et des soins.*

II. Le cancer du col de l'utérus et la nécessité de la biopsie dans la détection

1-Le Cancer du Col de l'Utérus

Le cancer du col de l'utérus est une maladie maligne qui se développe dans les cellules du col de l'utérus, la partie inférieure de l'utérus qui se connecte au vagin. Il est souvent causé par une infection persistante par certains types du virus du papillome humain

(VPH). Ce cancer est l'un des plus fréquents chez les femmes dans de nombreux pays, bien qu'il soit évitable grâce à des dépistages réguliers, comme le test de Papanicolaou (frottis cervical), et la vaccination contre le VPH.Les symptômes peuvent inclure des saignements vaginaux anormaux, des douleurs pelviennes ou des pertes vaginales inhabituelles. Cependant, dans ses premières étapes, le cancer du col de l'utérus peut être asymptomatique, d'où l'importance d'un suivi médical régulier pour une détection précoce. Un diagnostic précoce offre des chances de guérison plus élevées, et des traitements tels que la chirurgie, la radiothérapie et la chimiothérapie peuvent être utilisés selon le stade de la maladie.

2- La Nécessité de la Biopsie dans la Détection des Cancers

La biopsie est une procédure médicale essentielle dans le diagnostic du cancer. Elle consiste à prélever des échantillons de tissus ou de cellules du corps afin de les examiner au microscope. Cette analyse permet de déterminer la présence de cellules cancéreuses, leur type, leur degré d'agressivité et leur stade de développement.

L'importance de la biopsie dans la détection des cancers :

Confirmation du Diagnostic:

La biopsie permet de confirmer ou d'exclure la présence de cellules cancéreuses. Elle est souvent utilisée après des tests de dépistage comme les frottis cervicoutérins, qui peuvent indiquer des anomalies mais ne peuvent pas confirmer un cancer.

Identification du Type de Cancer :

En analysant les cellules prélevées, les pathologistes peuvent identifier le type de cancer et ses caractéristiques spécifiques. Cela inclut la taille, la forme et le degré d'agressivité des cellules cancéreuses.

Planification du Traitement :

Les informations obtenues grâce à la biopsie sont essentielles pour planifier le traitement. Elles aident les médecins à choisir les thérapies les plus appropriées en fonction du type et du stade du cancer.

Évaluation de l'Évolution de la Maladie :

La biopsie peut également être utilisée pour suivre l'évolution du cancer et évaluer l'efficacité des traitements en cours.

3. L'IA dans la détection du Cancer du Col de l'Utérus

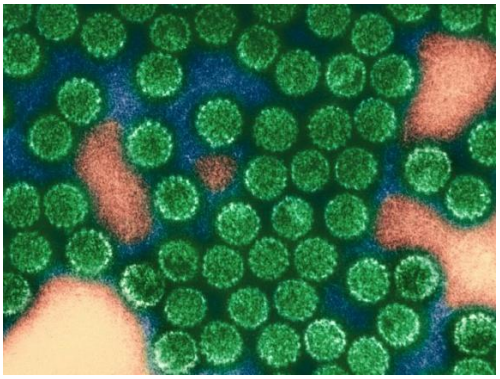
L'intelligence artificielle (IA) joue un rôle de plus en plus important dans la détection et le diagnostic du cancer, y compris le cancer du col de l'utérus. Les techniques d'IA, telles que l'apprentissage automatique (machine Learning) et l'apprentissage profond (Deep Learning), permettent d'analyser des volumes massifs de données médicales pour détecter des modèles subtils souvent invisibles à l'œil humain.

Dans la détection du cancer, l'IA est principalement utilisée pour analyser des images médicales (comme des radiographies, des échographies ou des frottis cervicaux), des résultats de tests diagnostiques et des données cliniques. Par exemple, des modèles d'IA peuvent être formés pour reconnaître les anomalies dans des images de cellules cervicales, ce qui permet une détection précoce des changements pathologiques indiquant un cancer. L'IA peut également être utilisée pour prédire le risque de développer un cancer en analysant des facteurs de risque tels que les antécédents médicaux, le mode de vie et les résultats des tests génétiques.

Les avantages de l'IA dans la détection du cancer incluent une plus grande précision, une analyse plus rapide des données et la réduction des erreurs humaines, ce qui permet aux médecins de poser des diagnostics plus fiables et de proposer des traitements plus personnalisés. De plus, l'IA peut également aider à suivre l'évolution de la maladie et à ajuster les traitements en fonction des réponses du patient.

Cependant, il est important de noter que l'IA ne remplace pas les médecins, mais les aide dans leur prise de décision. La combinaison de l'expertise humaine et des capacités analytiques de l'IA peut améliorer considérablement les résultats des patients.

En tant que débutants dans l'intelligence artificielle, nous avons choisi de nous limiter à une application simple pour prédire le cancer du col de l'utérus en utilisant des données tabulaires et un modèle CatBoost. Cette approche nous permet de nous concentrer sur les bases du machine Learning et de l'interprétation des résultats sans entrer dans des optimisations complexes.



Des techniques de détection innovantes du cancer du col de l'utérus

III. Outils Technologiques Utilisés dans le Projet

1.Les bibliothèques

Pour développer notre application de support à la décision médicale pour l'évaluation du risque de cancer du col de l'utérus, nous avons utilisé une combinaison de technologies avancées. Voici un aperçu des principaux outils technologiques intégrés, basés sur le fichier rééquipements.txt :

- *Stream lit*
  - ***Stream lit est un Framework open-source qui permet de créer des applications web interactives pour la machine Learning et la science des données. Il est particulièrement apprécié pour sa simplicité et sa capacité à transformer des scripts Python en applications web conviviales sans nécessiter de connaissances approfondies en développement web. Dans notre projet, Streamlit est utilisé pour développer l'interface utilisateur, permettant aux médecins de saisir les données des patients et de visualiser les prédictions et les explications.***
- *CatBoost*
  - ***CatBoost est un algorithme de gradient boosting développé par Yandex. Il est particulièrement efficace pour traiter des ensembles de données avec des caractéristiques catégorielles et offre des performances élevées avec***

peu de réglages. Nous utilisons CatBoost pour entraîner notre modèle de prédiction du risque de cancer du col de l'utérus, en tirant parti de sa robustesse et de sa capacité à gérer des données complexes.

- *SHAP (Shapley Additive explantations**)***
  - ***SHAP est une bibliothèque Python qui fournit des explications cohérentes et interprétables pour les prédictions des modèles de machine Learning. En utilisant les valeurs de Shapley, SHAP permet de comprendre l'importance de chaque caractéristique dans la prédiction d'un modèle. Dans notre projet, SHAP est utilisé pour expliquer les prédictions du modèle CatBoost, offrant ainsi une transparence essentielle pour les médecins et les patients.***
- *Pandas et NumPy*
  - ***Pandas et NumPy sont des bibliothèques fondamentales pour la manipulation et l'analyse des données en Python. Pandas offre des structures de données flexibles et puissantes, comme les DataFrames, tandis que NumPy fournit des outils pour travailler avec des tableaux multidimensionnels et des fonctions mathématiques. Ces bibliothèques sont utilisées pour le prétraitement des données, la gestion des entrées utilisateur et la préparation des données pour le modèle de machine Learning.***
- *Matplotlib*
  - ***Matplotlib est une bibliothèque de visualisation de données en Python qui permet de créer des graphiques statiques, animés et interactifs. Nous utilisons Matplotlib pour générer des visualisations des explications SHAP, aidant ainsi les utilisateurs à comprendre les facteurs influençant les prédictions du modèle.***
- *Joblib*
  - ***Joblib est une bibliothèque Python utilisée pour la sérialisation des objets, ce qui permet de sauvegarder et de charger des modèles de machine learning de manière efficace. Dans notre projet, Joblib est utilisé pour charger le modèle CatBoost pré-entraîné, garantissant ainsi une intégration fluide dans l'application Streamlit.***
- *Scikit-learn*
  - ***Scikit-learn est une bibliothèque de machine Learning en Python qui offre des outils simples et efficaces pour l'analyse des données et la modélisation prédictive. Nous utilisons Learning pour diverses tâches de prétraitement des données et pour l'évaluation des performances des modèles.***
- *Seaborn*
  - ***Seaborn est une bibliothèque de visualisation de données basée sur Matplotlib qui permet de créer des graphiques statistiques attrayants et***

informatifs. Nous utilisons Seaborn pour améliorer la visualisation des données et des résultats des modèles.

- *GitPython*
  - ***GitPython est une bibliothèque Python qui permet d'interagir avec des dépôts Git. Nous utilisons GitPython pour automatiser certaines tâches liées à la gestion du code source et à l'intégration continue.***
- *Trello*
  - ***Trello est un outil de gestion de projet visuel qui permet de suivre les tâches et de collaborer efficacement avec les membres de l'équipe. Nous utilisons Trello pour organiser les tâches, définir les responsabilités et suivre l'avancement du projet. En combinant ces outils technologiques, nous visons à développer une application de support à la décision médicale robuste, transparente et facile à utiliser, qui aidera les médecins à évaluer le risque de cancer du col de l'utérus de manière précise et explicable.***

2. Explication des algorithmes de machine Learning utilisés

Dans notre projet, nous avons utilisé plusieurs algorithmes de machine Learning pour évaluer le risque de cancer du col de l'utérus.

- *XGBoost*

- ***XGBoost (Extrême Gradient Boosting) est un algorithme de gradient boosting optimisé pour la vitesse et la performance. Voici ses principals caractéristiques et son fonctionnement :***

- ***Optimisation de la Vitesse : XGBoost est conçu pour être extrêmement rapide, grâce à des optimisations de calcul et de mémoire.***
- ***Régularisation : Il inclut des techniques de régularisation pour prévenir le surapprentissage.***
- ***Flexibilité : XGBoost peut être utilisé pour des tâches de classification et de régression, et il est très performant sur des ensembles de données divers.***
- ***Fonctionnement : XGBoost construit des modèles en ajoutant séquentiellement des arbres de décision faibles (learners) pour corriger les erreurs des modèles précédents. Chaque nouvel arbre est formé pour minimiser une fonction de perte en utilisant une technique d'optimisation par descente de gradient. Les prédictions finales sont obtenues en combinant les résultats de tous les arbres.***

- *CatBoost*

- ***CatBoost est un algorithme de gradient boosting développé par Yandex, particulièrement efficace pour traiter des ensembles de données avec des caractéristiques catégorielles. Voici ses principales caractéristiques et son fonctionnement :***

- ***Traitement des Données Catégorielles : CatBoost gère automatiquement les caractéristiques catégorielles, ce qui réduit le besoin de prétraitement des données.***
- ***Robustesse : Il est moins susceptible de surapprentissage (overfitting) grâce à ses techniques de régularisation.***
- ***Performance : CatBoost est rapide et peut être utilisé pour des tâches de classification et de régression avec des résultats précis.***
- ***Fonctionnement : CatBoost utilise le boosting ordonné pour réduire le surapprentissage et améliorer la précision des modèles. Il construit des arbres de décision symétriques en utilisant des permutations aléatoires et une optimisation basée sur le gradient. Les arbres sont formés pour minimiser la perte en se concentrant sur les régions de l'espace des caractéristiques ayant le plus grand impact sur la fonction de perte.***

En termes plus techniques, les différentes variantes du *boosting* partagent toutes trois caractéristiques communes:

- Ils visent à **trouver une approximation** $\hat{F}$ d'une fonction inconnue $F^* : \mathbf{x} \mapsto y$ à partir d'un ensemble d'entraînement $(y_i, \mathbf{x_i})_{i=1,\ldots,n}$;
- Ils supposent que la fonction F^* peut être approchée par une **somme pondérée de modèles simples** f de paramètres θ:

$$F(\mathbf{x}) = \sum_{m=1}^{M} \beta_m f(\mathbf{x}, \theta_m)$$

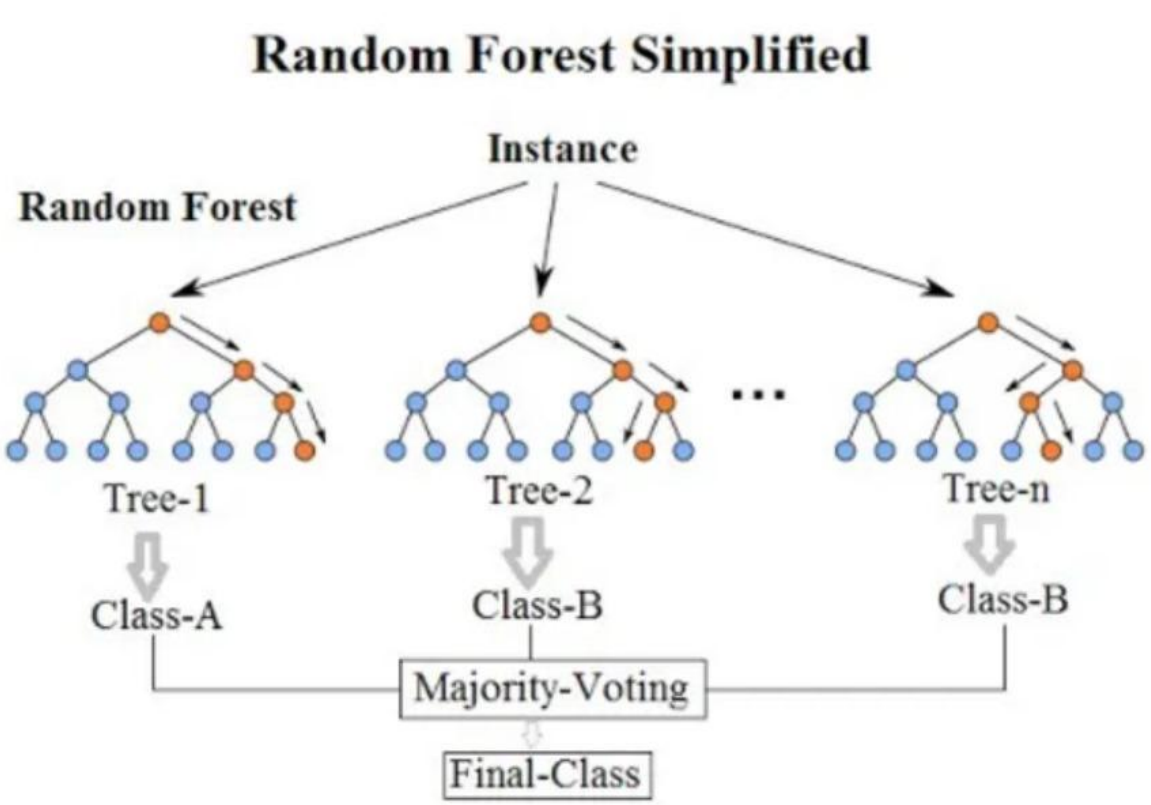
- Ils reposent sur une **modélisation additive par étapes** (*forward stagewise additive modeling*), qui décompose l'entraînement de ce modèle complexe en une **séquence d'entraînements de petits modèles**. Chaque étape de l'entraînement cherche le modèle simple f qui améliore la puissance prédictive du modèle complet, sans modifier les modèles précédents, puis l'ajoute de façon incrémentale à ces derniers:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \hat{\beta}_m f(\mathbf{x}_i, \hat{\theta}_\mathbf{m})$$

Equation générique des méthodes Catboost et XGboost

- *Random Forest*
- ***Le Random Forest est un algorithme d'ensemble qui utilise plusieurs arbres de décision pour améliorer la précision des prédictions.***
- ***Ensemble de Modèles : Il construit plusieurs arbres de décision à partir de sousensembles aléatoires des données et combine leurs résultats pour obtenir une prédiction finale.***
- ***Réduction du Surapprentissage : En utilisant plusieurs arbres, le Random Forest réduit le risque de surapprentissage par rapport à un arbre de décision unique.***
- ***Interprétabilité : Bien que moins interprétable que les arbres de décision simples, il offre une bonne compréhension des caractéristiques importantes grâce à l'analyse de l'importance des variables.***
- ***Fonctionnement : Le Random Forest utilise la méthode de bagging (bootstrap aggregating) pour créer des sous-ensembles aléatoires de données et des sousensembles aléatoires de caractéristiques pour chaque arbre. Chaque arbre est***

formé indépendamment et les prédictions finales sont obtenues en agrégeant les résultats de tous les arbres (par vote majoritaire pour la classification ou par moyenne pour la régression).



Modèle Random Forest

- *Support Vector Machines (SVM)*
- *Les SVM sont des algorithmes de classification qui cherchent à trouver l'hyperplan optimal séparant les différentes classes dans l'espace des caractéristiques.*
- *Hyperplan Optimal : Les SVM maximisent la marge entre les classes, ce qui les rend efficaces pour les tâches de classification.*
- *Kernel Trick : Ils peuvent utiliser des fonctions de noyau pour transformer les données et trouver des séparations non linéaires.*
- *Robustesse : Les SVM sont robustes aux surapprentissages et fonctionnent bien avec des ensembles de données de haute dimension.*

- ***Fonctionnement : Les SVM cherchent à maximiser la marge entre les points de données les plus proches de l'hyperplan (appelés vecteurs de support) et l'hyperplan lui-même. En utilisant des fonctions de noyau, les SVM peuvent transformer les données dans un espace de caractéristiques de plus haute dimension où un hyperplan linéaire peut séparer les classes.***
- ***En utilisant ces algorithmes, nous avons pu développer un modèle robuste et explicable pour évaluer le risque de cancer du col de l'utérus, en tirant parti des forces de chaque méthode pour améliorer la précision et la transparence des prédictions.***

3. Méthode smote

La méthode SMOTE (Synthetic Minority Over-sampling Technique) est une technique de suréchantillonnage utilisée pour traiter les ensembles de données déséquilibrés en machine learning. Dans les ensembles de données déséquilibrés, il y a souvent beaucoup plus d'exemples de la classe majoritaire que de la classe minoritaire. Cela peut poser des problèmes pour les algorithmes de machine learning, qui peuvent avoir du mal à apprendre des caractéristiques pertinentes de la classe minoritaire. SMOTE est utilisé pour augmenter le nombre d'exemples de la classe minoritaire, ce qui aide à équilibrer la distribution des classes et à améliorer la performance des modèles.

IV. Explication du nettoyage de la data

1. Visualisation

Nous avons commencé par importer les bibliothèques dont on aura besoin pour le nettoyage.

```
from ucimlrepo import fetch_ucirepo
import pandas as pd
import plotly.express as px
import seaborn as sns
import plotly.subplots as sp
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import shap
```

+ Code + Marquage

Importation des bibliothèques

Après cela, nous chargeons la data dans un dataframe

```
cervical_cancer_risk_factors = fetch_ucirepo(id=383)
data=cervical_cancer_risk_factors.data.features
```

Chargement du *dataframe*

Nous avons par la suite essayer de visualiser les premières lignes de notre data pour voir à quoi cela ressemble.

```
df=data.copy()
df.head()
```

Python

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives	Hormonal Contraceptives (years)	IUD
0	18	4.0	15.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1	15	1.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	34	1.0	NaN	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	52	5.0	16.0	4.0	1.0	37.0	37.0	1.0	3.0	0.0
4	46	3.0	21.0	4.0	0.0	0.0	0.0	1.0	15.0	0.0

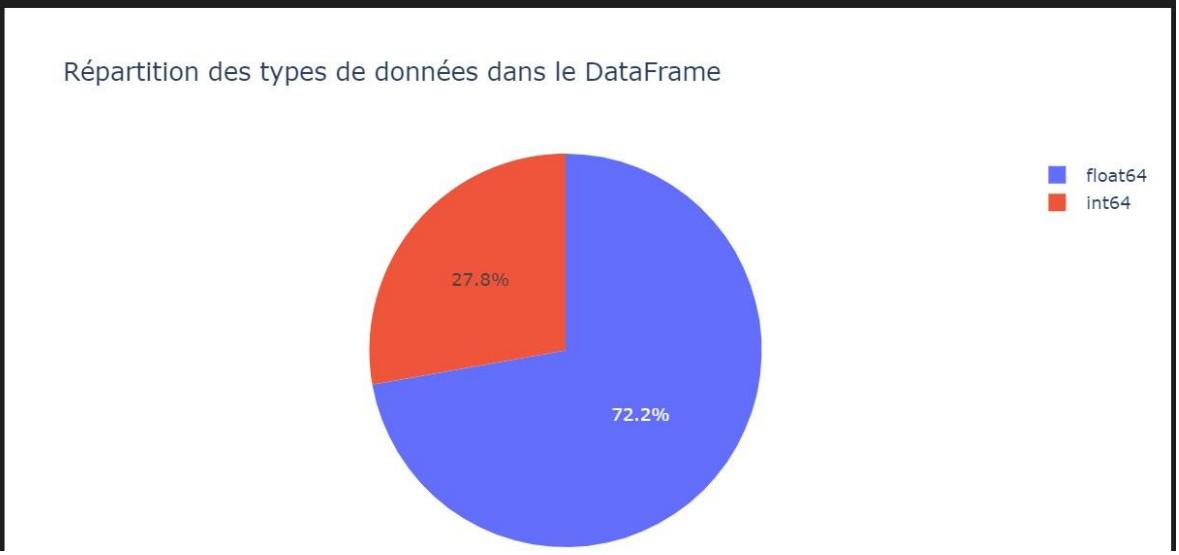
5 rows × 36 columns

les premières lignes de notre data

Afin d'analyser la structure du jeu de données, nous avons créé une visualisation sous forme de diagramme circulaire (ou camembert) pour représenter la répartition des types de données dans le DataFrame. Cette approche permet de mieux comprendre la composition des variables, notamment le nombre de colonnes de type numérique, catégoriel, ou autre.

```
    values=df.dtypes.value_counts().values,
    title="Répartition des types de données dans le DataFrame"
)
fig.show()
```

[7] ✓ 1.6s Pytho



Repartition de la data en int et float

2. Gestion des colonnes avec beaucoup de valeurs manquantes

Pour évaluer la qualité des données et identifier les colonnes avec des valeurs manquantes, nous avons calculé le ratio des données manquantes pour chaque colonne du DataFrame

```
#Calcul su ratio des données manquantes par colones
(df.isna().sum().sort_values(ascending=False))/(df.shape[0])
```

✓ 0.0s

```
STDs: Time since last diagnosis       0.917249
STDs: Time since first diagnosis      0.917249
IUD                                   0.136364
IUD (years)                           0.136364
Hormonal Contraceptives               0.125874
Hormonal Contraceptives (years)       0.125874
STDs:pelvic inflammatory disease      0.122378
STDs:vulvo-perineal condylomatosis    0.122378
```

Visualisation du pourcentage de quelques valeur manquants

Le calcul du ratio des valeurs manquantes par colonne permet d'identifier rapidement les variables qui nécessitent un nettoyage ou un traitement particulier. Cette étape est cruciale, car les données manquantes peuvent fausser les résultats des analyses ou des modèles de Machine Learning.

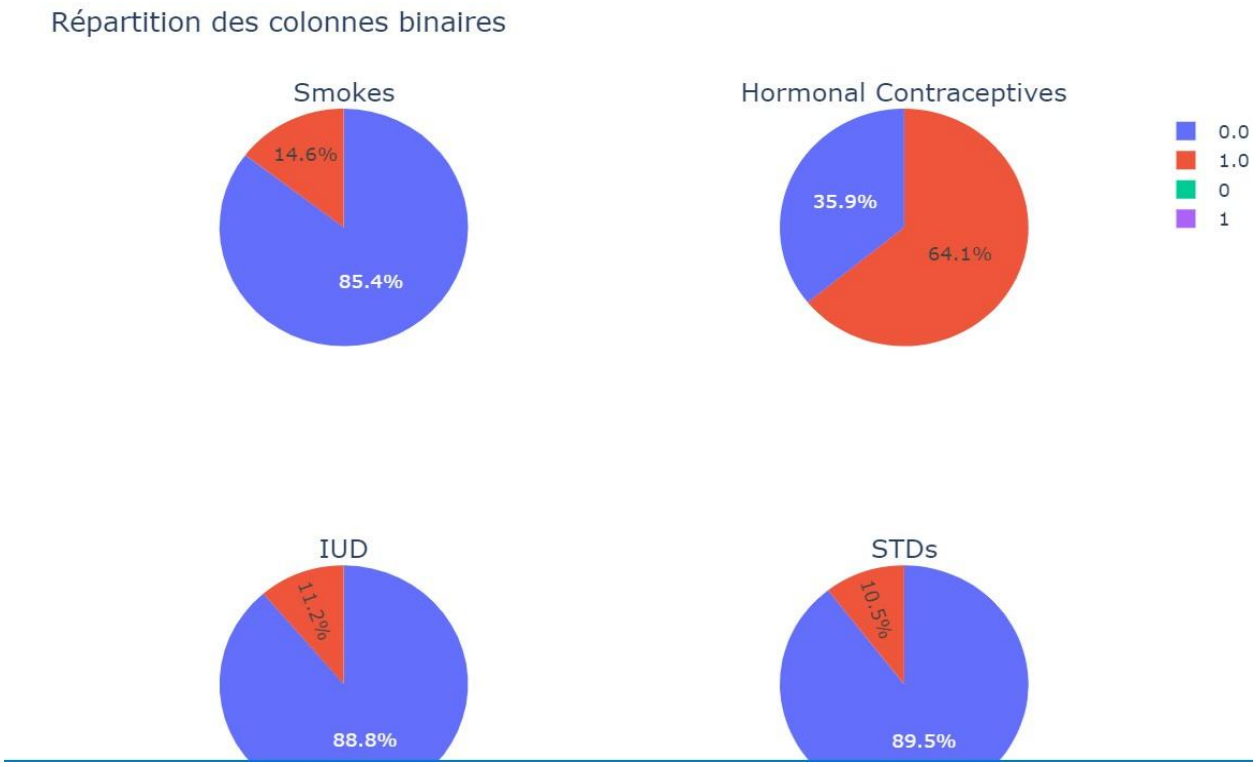
Suite à ces résultats, nous avons décidé de supprimer les colonnes STDs: Time since first diagnosis et STDs: Time since last diagnosis en raison de leur ratio élevé de valeurs manquantes (0,91), rendant leur imputation difficile et incertaine. De plus, ces informations sont peu représentatives pour l'analyse, car elles ne sont pas disponibles pour tous les individus. Par conséquent, leur suppression permet de garantir la fiabilité des analyses et des modèles.

3. Les variables binaires

Nous avons par la suite identifier et lister toutes les colonnes du DataFrame df qui contiennent des variables binaires, c'est-à-dire des colonnes ayant deux valeurs distinctes (ou moins). Ces variables binaires peuvent représenter des catégories telles que oui/non, vrai/faux, présent/absent,etc.

Les variables binaires sont souvent utilisées comme des indicateurs dans les modèles de

Machine Learning, car elles peuvent être facilement converties en variables numériques (0 ou 1). Cette étape permet de préparer les colonnes appropriées pour un traitement ultérieur, comme l'encodage ou l'analyse de leur impact sur les prédictions.



Visualisation de quelques variables binaires

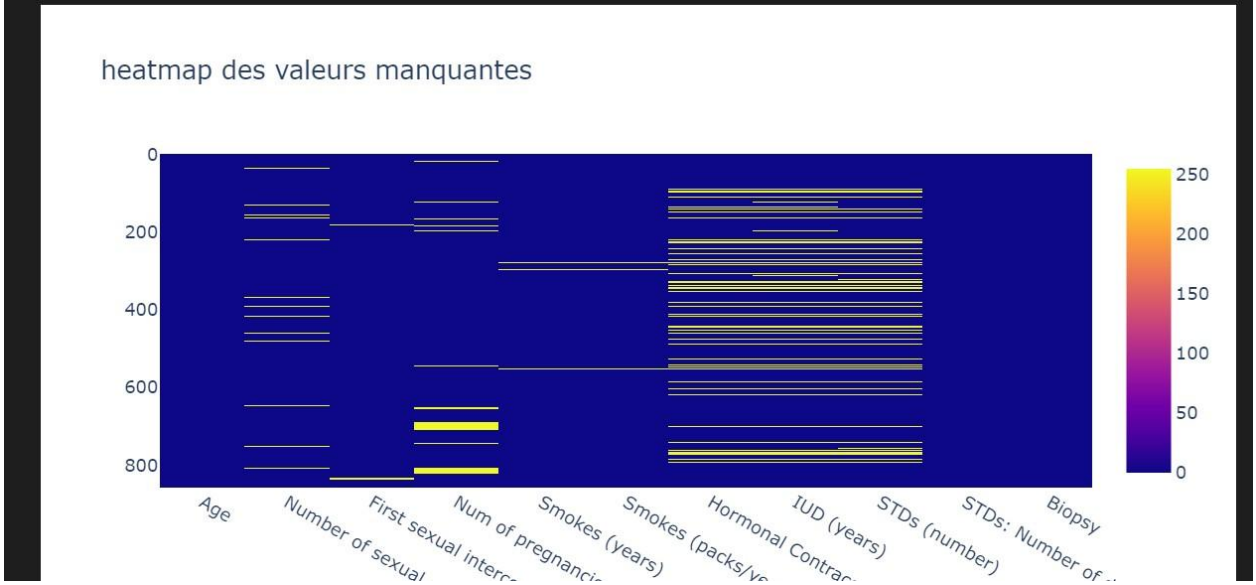
On peut constater apres traçage que beaucoup de variable ne sont pas necessaire ,on va tout supprimer sauf la biopsy, colonne cible.

4-Distribution des valeurs manquantes

Une heatmap a été générée pour visualiser la répartition des valeurs manquantes dans le DataFrame. Cette visualisation permet d'identifier rapidement les colonnes et lignes présentant des données manquantes, facilitant ainsi les décisions sur le traitement à adopter (suppression ou imputation des valeurs manquantes).

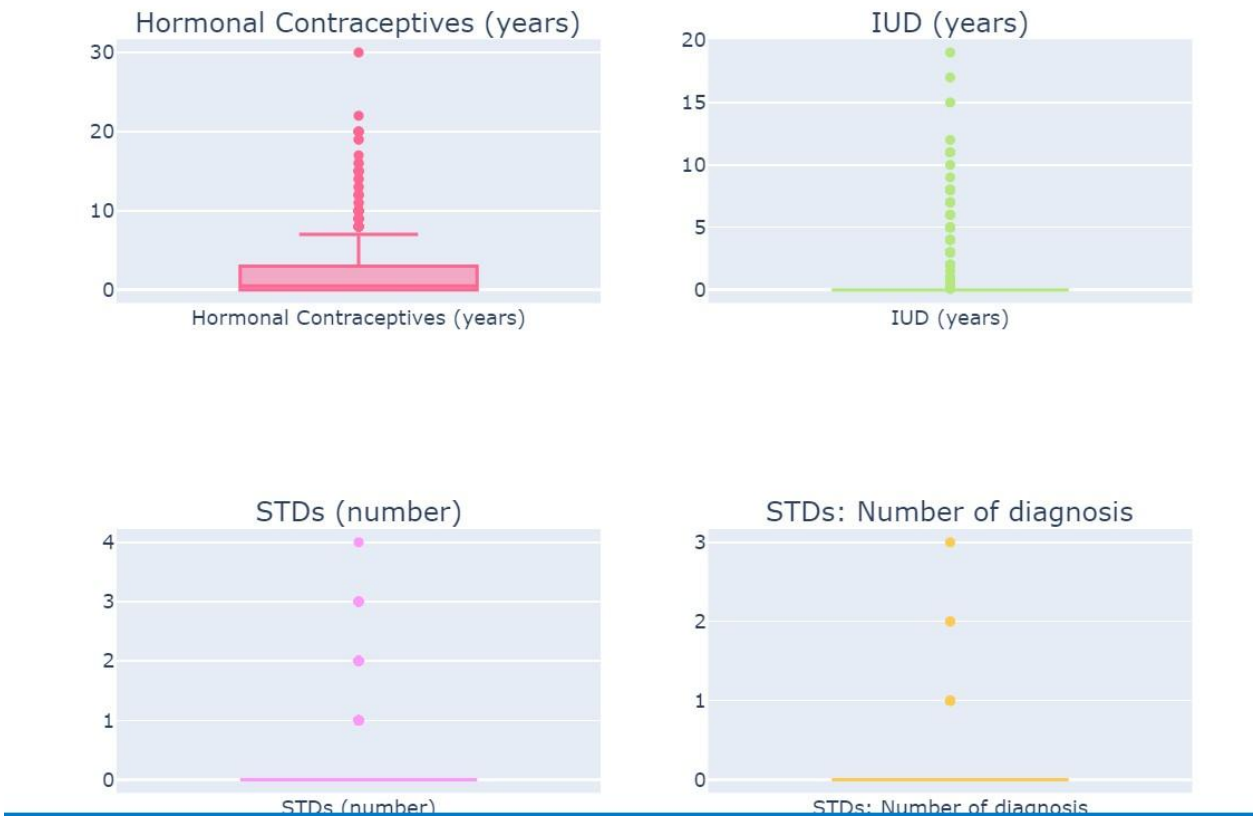
```
#visualisation des valeur manquantes
px.imshow(df.isnull(),title='heatmap des valeurs manquantes')
#on constate à travers cette heatmpa la distribution des valeur manquantes
```

✓ 0.2s Python



Une heatmap

Des boxplots ont été tracés pour les 11 colonnes numériques du DataFrame afin d'identifier les outliers (valeurs aberrantes). Ces graphiques permettent de visualiser la distribution des données, les quartiles ainsi que les points extrêmes, facilitant ainsi la détection des anomalies dans chaque colonne.



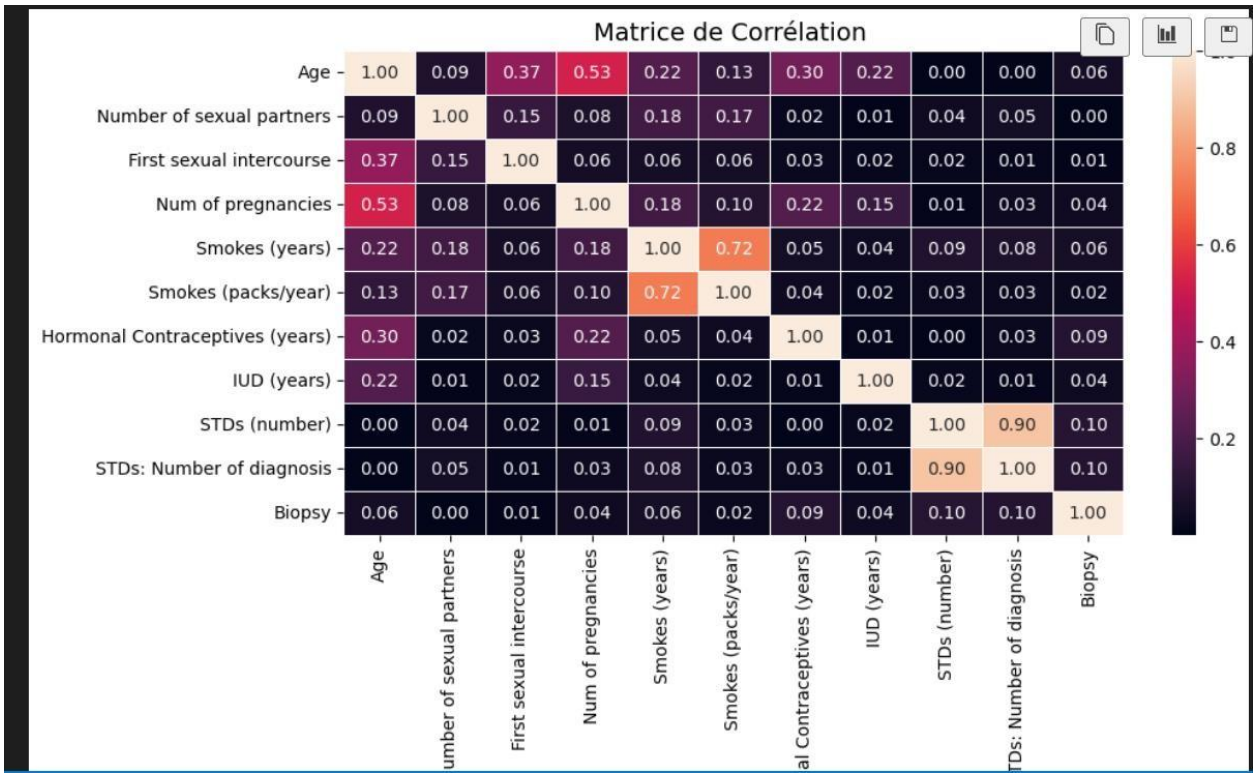
Visualisation de quelques boxplots

Les valeurs manquantes dans le DataFrame ont été remplies avec la médiane de chaque colonne car la médiane est robuste aux outliers, mieux adaptée aux distributions asymétriques, et préserve la répartition des données sans être influencée par des valeurs extrêmes. Cette méthode est utilisée pour traiter les données manquantes tout en préservant la distribution des données, en particulier pour les variables numériques, afin d'éviter d'introduire des biais dans l'analyse.

5-Matrice de corrélation

Une matrice de corrélation a été générée pour examiner les relations entre les variables numériques du DataFrame. Elle met en évidence les corrélations entre les différentes variables, montrant que seules quelques variables présentent des corrélations

significatives. Cela permet de mieux comprendre les dépendances entre les caractéristiques avant de procéder à l'analyse plus approfondie ou à la modélisation.



La matrice de corrélation

Les variables corrélées n'ont pas été supprimées à ce stade, car l'objectif était d'explorer les relations entre les variables sans perdre d'informations. La suppression des variables corrélées peut être réalisée plus tard, selon les besoins du modèle, notamment en utilisant des techniques comme la réduction de dimension. De plus, certains modèles peuvent gérer la multicolinéarité sans nécessiter la suppression des variables.

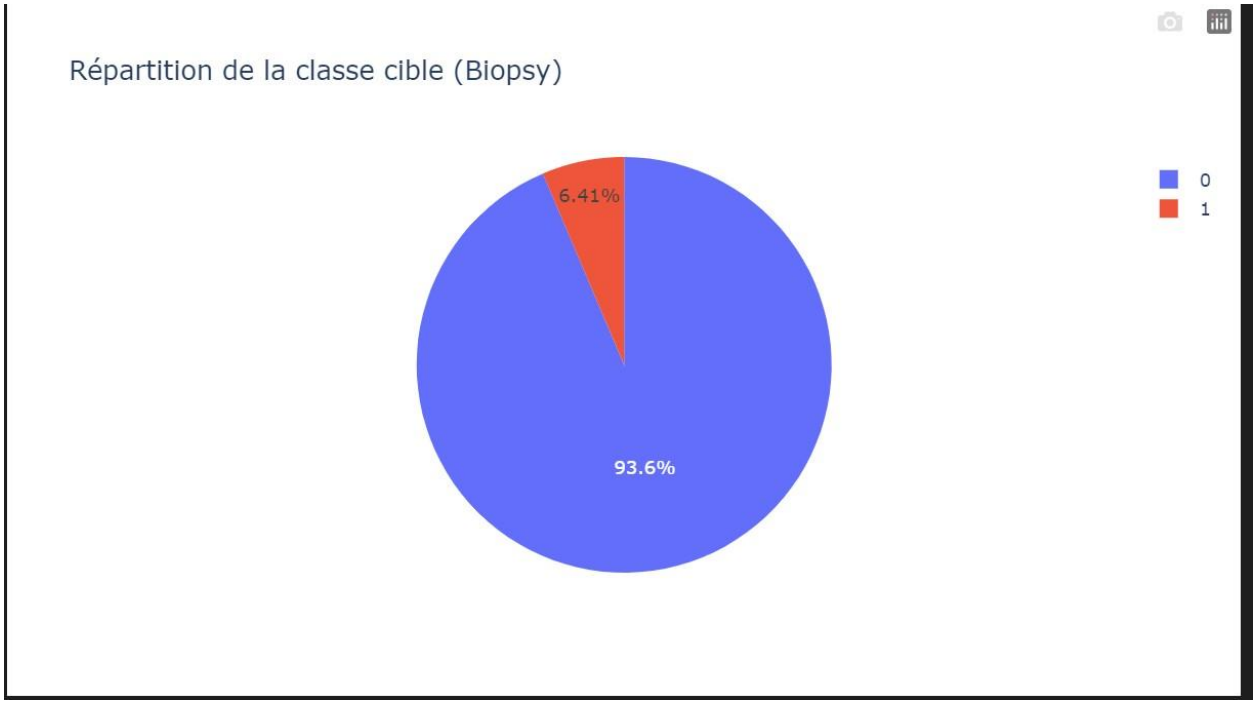
V. Models et précision graphique

Nous avons généré une fonction evaluate_classification_model_plotly qui évalue un modèle de classification en générant plusieurs visualisations et rapports. Il affiche les matrices de confusion pour les ensembles d'entraînement et de test, trace les courbes

ROC avec les valeurs AUC pour chaque ensemble, et présente les métriques de classification (précision, rappel, F1-score). Ces éléments permettent d'analyser la performance du modèle de manière complète et visuelle.

1. Visualisation de la classe biopsy

Un diagramme circulaire a été créé pour visualiser la répartition de la classe cible "Biopsy".
Cette visualisation permet de vérifier si la variable cible présente un déséquilibre, c'est-à-dire si certaines classes sont sous-représentées par rapport à d'autres. Cela est important pour adapter les techniques de modélisation en fonction du déséquilibre éventuel.



IL y a plus de zéro que de un dans la biopsy

Les données ont été divisées en ensembles d'entraînement et de test à l'aide de la fonction train_test_split de Scikit-learn. La colonne "Biopsy" a été séparée comme variable cible (y), et les autres colonnes comme variables explicatives (X). L'option stratify=y assure que la répartition des classes dans les ensembles d'entraînement et de test reste équilibrée. La division a été effectuée de manière aléatoire avec une graine définie (random_state=47) pour garantir la reproductibilité des résultats.

Lors de l'entraînement des différents modèles sur un ensemble de données contenant un déséquilibre de classes (par exemple, une classe cible "Biopsy" fortement déséquilibrée), nous avons obtenue de mauvais résultats. Les modèles tels que XGBoost, Cataboost et Random Forest ont eu tous tendance à prédire majoritairement la classe majoritaire, car il est biaisé par l'absence de données suffisantes de la classe minoritaire. Cela a conduit à des performances dégradées, notamment une faible précision et un score F1 faible pour la classe sousreprésentée. En conséquence, les métriques de performance, telles que la précision, le rappel et la courbe ROC, peuvent ne pas refléter correctement la capacité du modèle à distinguer entre les différentes classes, car le modèle favorise la classe majoritaire. Pour remédier à cela, il est essentiel de rééchantillonner les données par l'utilisation de techniques spécifiques telles que le SMOTE (Synthetic Minority Over-sampling Technique).

2. Gestion du Déséquilibre des Classes avec SMOTE

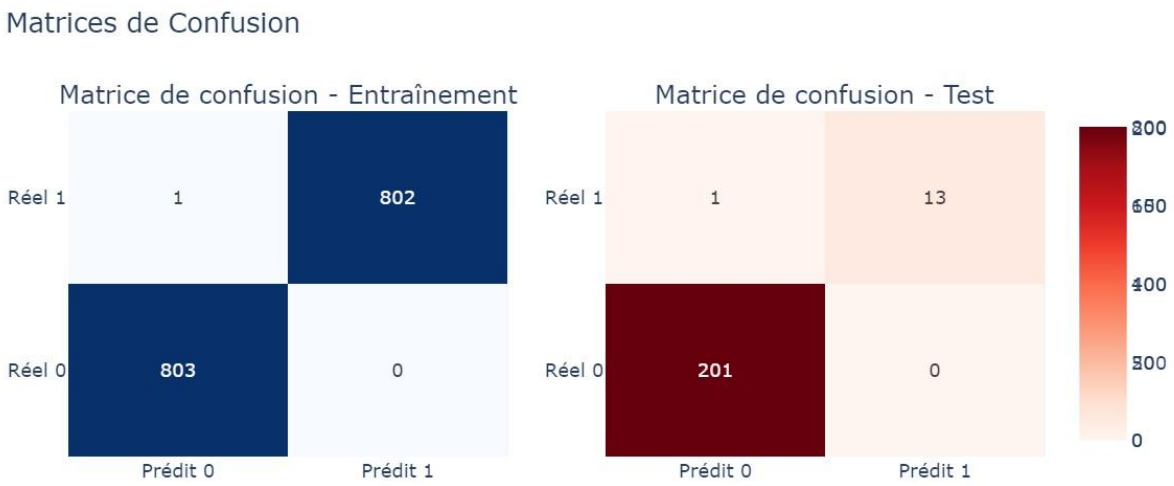
Afin de traiter le déséquilibre des classes dans l'ensemble d'entraînement, la méthode SMOTE (Synthetic Minority Over-sampling Technique) a été utilisée. SMOTE génère de nouvelles instances synthétiques pour la classe minoritaire en utilisant les voisins les plus proches (ici, k_neighbors=3). Cette approche permet d'équilibrer les classes et de rendre le modèle plus sensible aux classes sous-représentées, réduisant ainsi le biais vers la classe majoritaire.

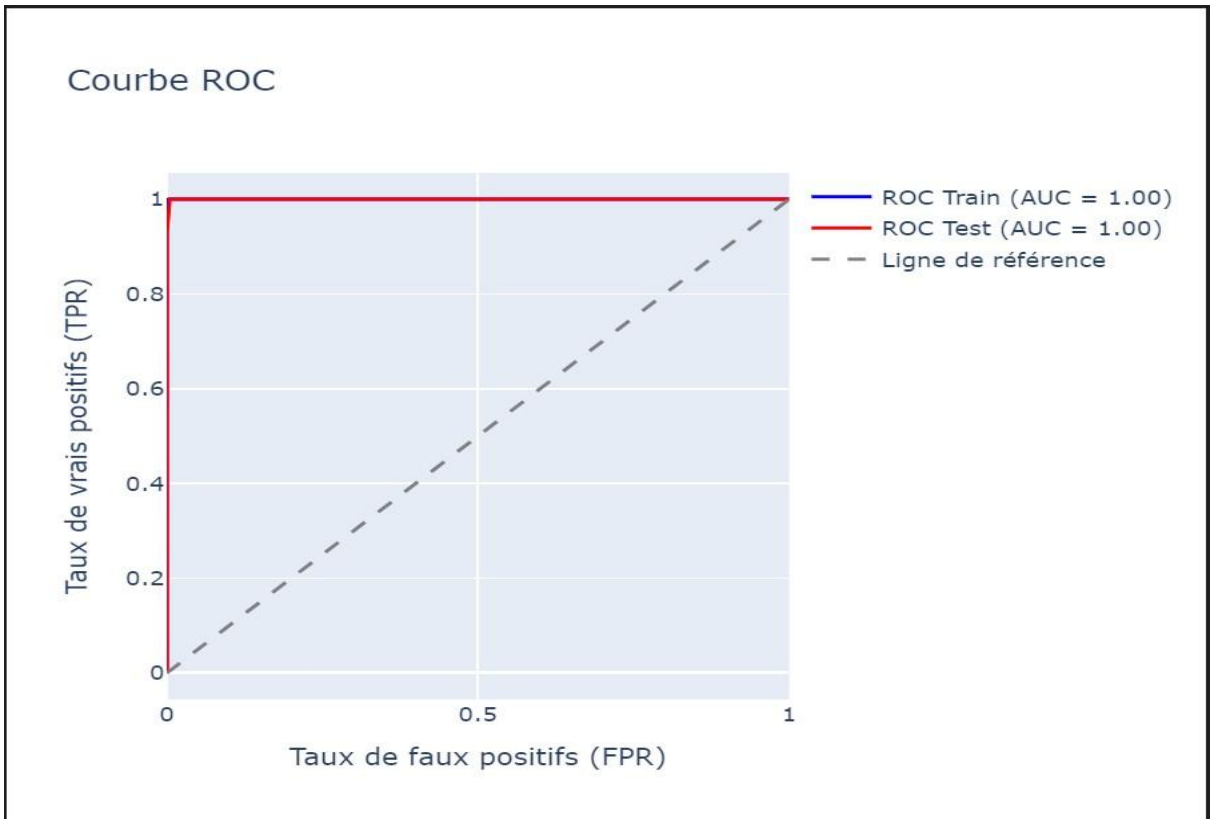
3. Standardisation des Données

Les données ont été standardisées à l'aide de la classe StandardScaler de Scikit-learn. Cette étape consiste à ajuster les données d'entraînement (x_train) et de test (x_test) de manière à ce qu'elles aient une moyenne de 0 et un écart-type de 1. La standardisation est cruciale pour les modèles sensibles à l'échelle des caractéristiques, comme la régression logistique, les SVM, ou les réseaux de neurones, afin d'éviter que certaines variables dominent le modèle en raison de différences d'échelle.

4.Entraînement du Modèle RandomForest

Un modèle de forêt aléatoire (Random Forest) a été entraîné sur les données d'entraînement après standardisation. Le modèle utilise 150 arbres (n_estimators=150), le critère entropy pour la mesure de l'impureté et une profondeur maximale de 20 (max_depth=20) pour contrôler la complexité de l'arbre. Après l'entraînement, la fonction evaluate_classification_model_plotly a été utilisée pour évaluer la performance du modèle sur les ensembles d'entraînement et de test, en affichant des visualisations des matrices de confusion, des courbes ROC, et des métriques de classification.


Matrices de Confusion
Matrice de confusion - Entraînement
Réel 1
Réel 0
1
802
803
0
Prédit 0
Prédit 1
Matrice de confusion - Test
Réel 1
Réel 0
1
13
201
0
Prédit 0
Prédit 1
800
600
400
200
0


Courbe ROC
ROC Train (AUC = 1.00)
ROC Test (AUC = 1.00)
Ligne de référence
Taux de vrais positifs (TPR)
Taux de faux positifs (FPR)
0
0.2
0.4
0.6
0.8
1
0
0.5
1

```
Classification Report - Entraînement:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       803
           1       1.00      1.00      1.00       803

    accuracy                           1.00      1606
   macro avg       1.00      1.00      1.00      1606
weighted avg       1.00      1.00      1.00      1606

Classification Report - Test:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       201
           1       1.00      0.93      0.96        14

    accuracy                           1.00       215
   macro avg       1.00      0.96      0.98       215
weighted avg       1.00      1.00      1.00       215
```

Les résultats du Modèle Random Forest

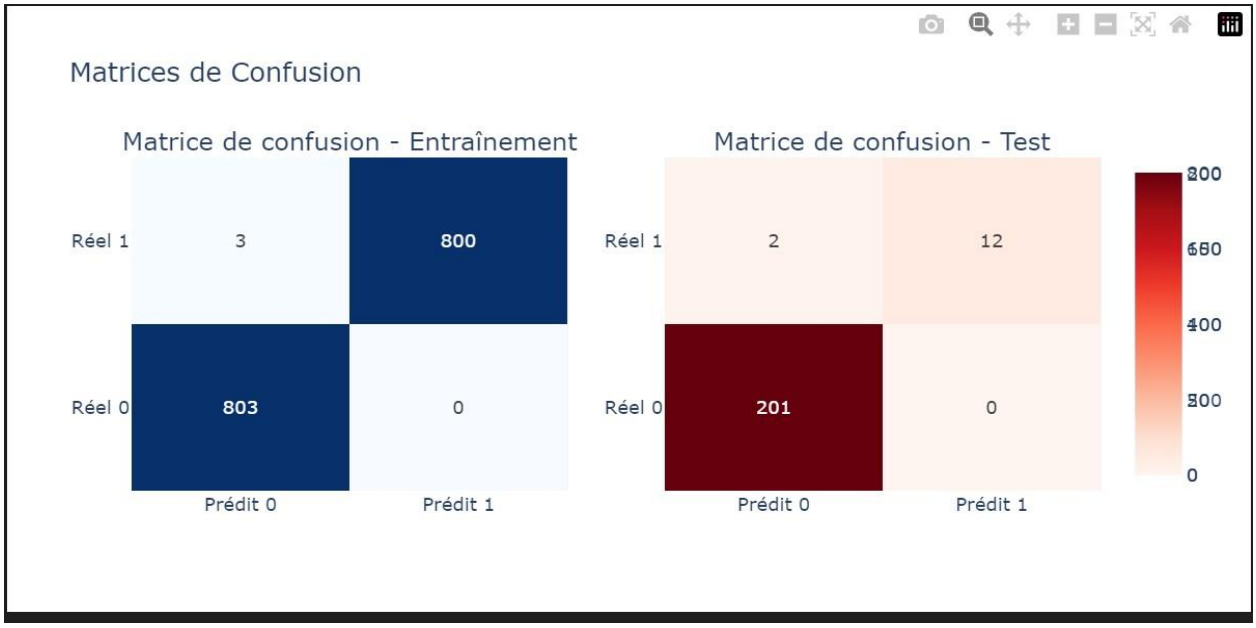
Le rapport de classification pour l'ensemble d'entraînement montre des résultats parfaits, avec des précisions, rappels et scores F1 de 1.00 pour les deux classes (0 et 1), ce qui suggère un modèle parfaitement ajusté aux données d'entraînement.

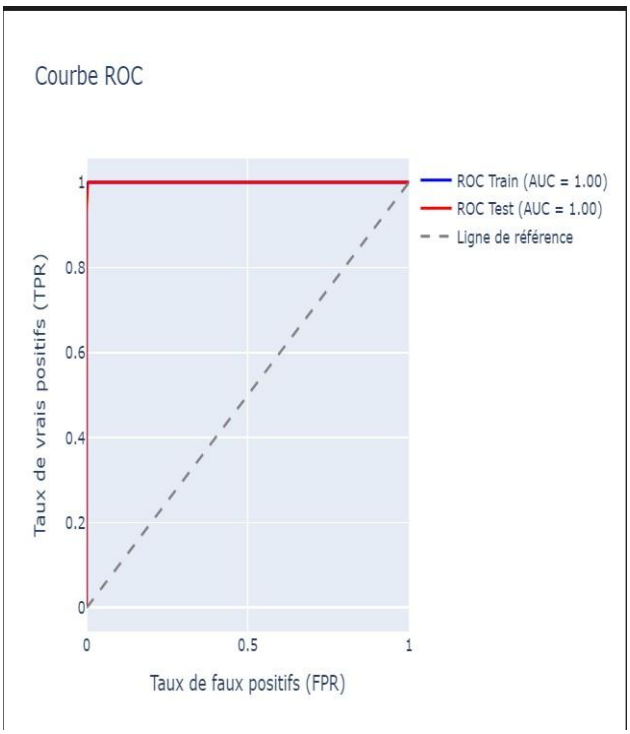
Cependant, les résultats pour l'ensemble de test montrent une légère dégradation de la performance pour la classe 1 (rappel de 0.93 et F1-score de 0.96). Bien que l'exactitude globale du modèle soit de 1.00, cela suggère un possible déséquilibre de classes (la classe 1 étant sousreprésentée), ce qui peut avoir conduit à une légère perte de performance sur cette classe minoritaire. Le modèle prédit parfaitement la classe majoritaire, mais la classe minoritaire pourrait nécessiter une attention supplémentaire pour améliorer la généralisation du modèle.

5. Entraînement du Modèle XGBoost

Un modèle XGBoost a été entraîné sur les données d'entraînement. Ce modèle utilise 100 arbres (n_estimators=100), une profondeur maximale de 15 (max_depth=15), et un taux

d'apprentissage de 0.1 (learning_rate=0.1). L'objectif de la classification est défini comme 'binary:logistic', adapté à un problème de classification binaire. Après l'entraînement, la fonction evaluate_classification_model_plotly a été utilisée pour évaluer la performance du modèle en générant des visualisations telles que les matrices de confusion, les courbes ROC et les métriques de classification.






```
Classification Report - Entraînement:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       803
           1       1.00      1.00      1.00       803

    accuracy                           1.00      1606
   macro avg       1.00      1.00      1.00      1606
weighted avg       1.00      1.00      1.00      1606

Classification Report - Test:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       201
           1       1.00      0.86      0.92        14

    accuracy                           0.99       215
   macro avg       1.00      0.93      0.96       215
weighted avg       0.99      0.99      0.99       215
```

Les résultats du Modèle XGBoost

Le rapport de classification pour l'ensemble d'entraînement montre des résultats parfaits, avec des précisions, rappels et scores F1 de 1.00 pour les deux classes (0 et 1). Cela indique que le modèle s'ajuste très bien aux données d'entraînement, sans signes de sur-apprentissage évident.

Pour l'ensemble de test, bien que l'exactitude globale soit de 0.99, il y a une légère baisse de performance pour la classe 1. Le modèle obtient un rappel de 0.86 et un score F1 de 0.92 pour cette classe, ce qui suggère que le modèle a des difficultés à prédire correctement la classe minoritaire (classe 1), possiblement en raison de son déséquilibre dans les données. Cependant, la classe 0 reste bien prédite, avec des résultats proches de 1.00.

Cela met en évidence un comportement similaire à celui observé avec le modèle Random Forest, où l'équilibre des classes reste un défi majeur pour prédire correctement les classes sous-représentées.

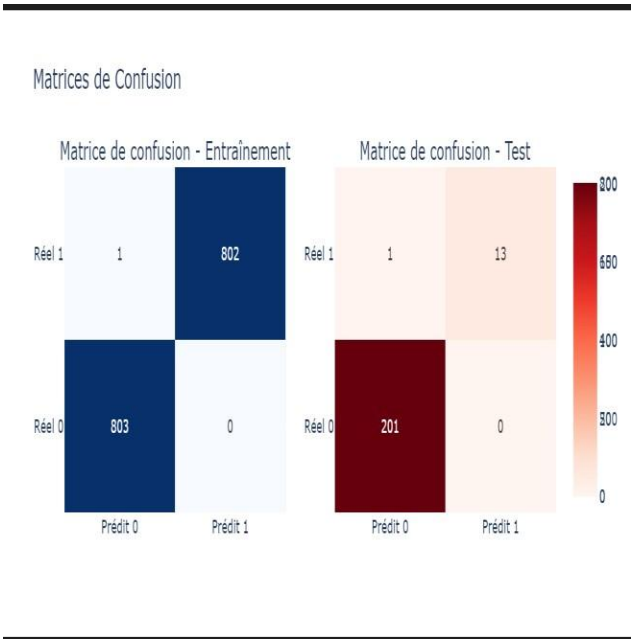
6. Entraînement du Modèle CatBoost

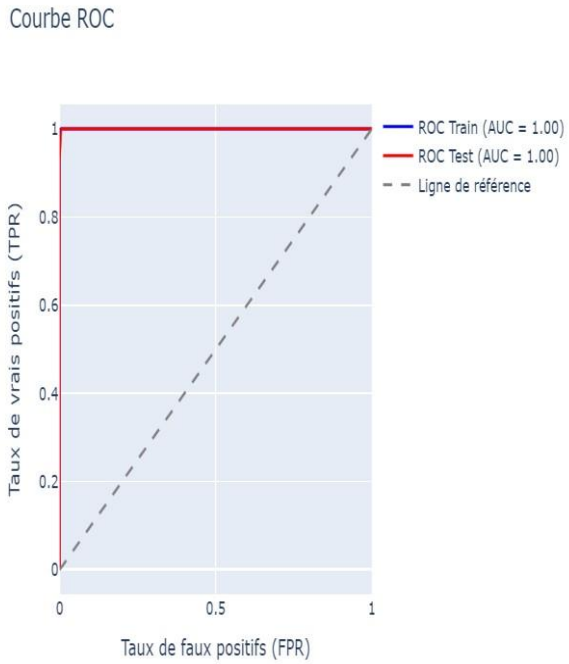
Un modèle CatBoost a été entraîné sur les données d'entraînement avec 500 itérations (iterations=500), un taux d'apprentissage de 0.1 (learning_rate=0.1), et une fonction de perte Logloss adaptée à un problème de classification binaire. Le modèle utilise également la métrique F1 pour évaluer ses performances pendant l'entraînement. Après l'entraînement, la fonction evaluate_classification_model_plotly a été utilisée pour évaluer la performance du modèle, en affichant des visualisations telles que les matrices de confusion, les courbes ROC et les métriques de classification.

```
10:     learn: 0.3825570        total: 278ms    remaining: 12.4s
11:     learn: 0.3648276        total: 294ms    remaining: 11.9s
12:     learn: 0.3489746        total: 309ms    remaining: 11.6s
13:     learn: 0.3418014        total: 321ms    remaining: 11.2s
14:     learn: 0.3269936        total: 334ms    remaining: 10.8s
15:     learn: 0.3175776        total: 346ms    remaining: 10.5s
16:     learn: 0.3108892        total: 359ms    remaining: 10.2s
17:     learn: 0.2998744        total: 372ms    remaining: 9.97s
18:     learn: 0.2913675        total: 385ms    remaining: 9.75s
19:     learn: 0.2859126        total: 397ms    remaining: 9.53s
20:     learn: 0.2813651        total: 409ms    remaining: 9.32s
21:     learn: 0.2764419        total: 422ms    remaining: 9.18s
22:     learn: 0.2676243        total: 436ms    remaining: 9.04s
23:     learn: 0.2579654        total: 448ms    remaining: 8.89s
24:     learn: 0.2517346        total: 457ms    remaining: 8.68s
...
496:    learn: 0.0136181        total: 4.35s    remaining: 26.3ms
497:    learn: 0.0135495        total: 4.36s    remaining: 17.5ms
498:    learn: 0.0134528        total: 4.36s    remaining: 8.73ms
499:    learn: 0.0134199        total: 4.36s    remaining: 0us
```

entraînement du modèle CatBoost

L'image montre l'entraînement du modèle CatBoost sur 500 itérations. À chaque itération, la fonction de perte Logloss diminue, indiquant que le modèle améliore progressivement ses prédictions. Le temps d'entraînement est également suivi, avec un temps de calcul total de 4.36 secondes pour les 500 itérations.






```
Classification Report - Entraînement:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       803
           1       1.00      1.00      1.00       803

    accuracy                           1.00      1606
   macro avg       1.00      1.00      1.00      1606
weighted avg       1.00      1.00      1.00      1606

Classification Report - Test:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       201
           1       1.00      0.93      0.96        14

    accuracy                           1.00       215
   macro avg       1.00      0.96      0.98       215
weighted avg       1.00      1.00      1.00       215
```

Les résultats du Modèle Catboost

Le modèle CatBoostClassifier obtient une précision parfaite sur l'ensemble d'entraînement, ce qui peut indiquer un surapprentissage. Sur l'ensemble de test, la performance reste très élevée, mais avec une légère baisse du rappel pour la classe minoritaire (1), ce qui suggère que le modèle pourrait encore être optimisé pour mieux généraliser aux nouveaux échantillons.

Içi, nous analysons l'importance des variables utilisées par le modèle CatBoost, sélectionné comme le meilleur modèle. En évaluant l'importance des caractéristiques, on identifie celles qui contribuent le plus aux prédictions, ce qui peut aider à optimiser le modèle ou à interpréter les facteurs influents dans le diagnostic.

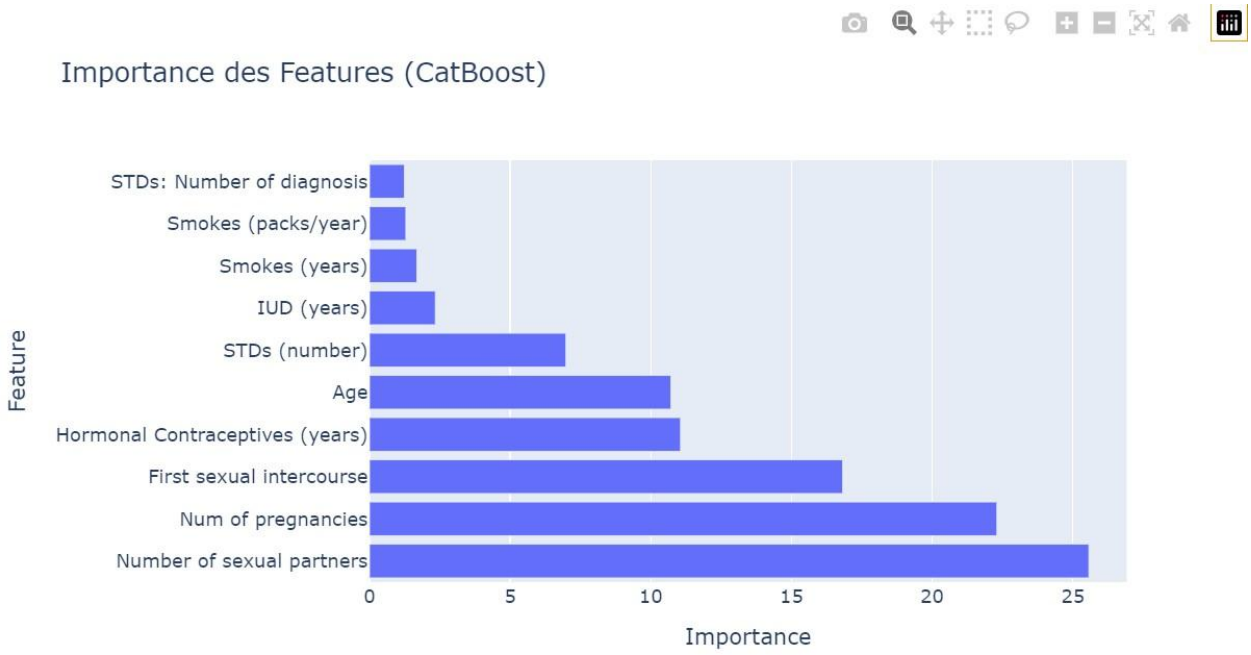


Figure illustrative de l'importance des features

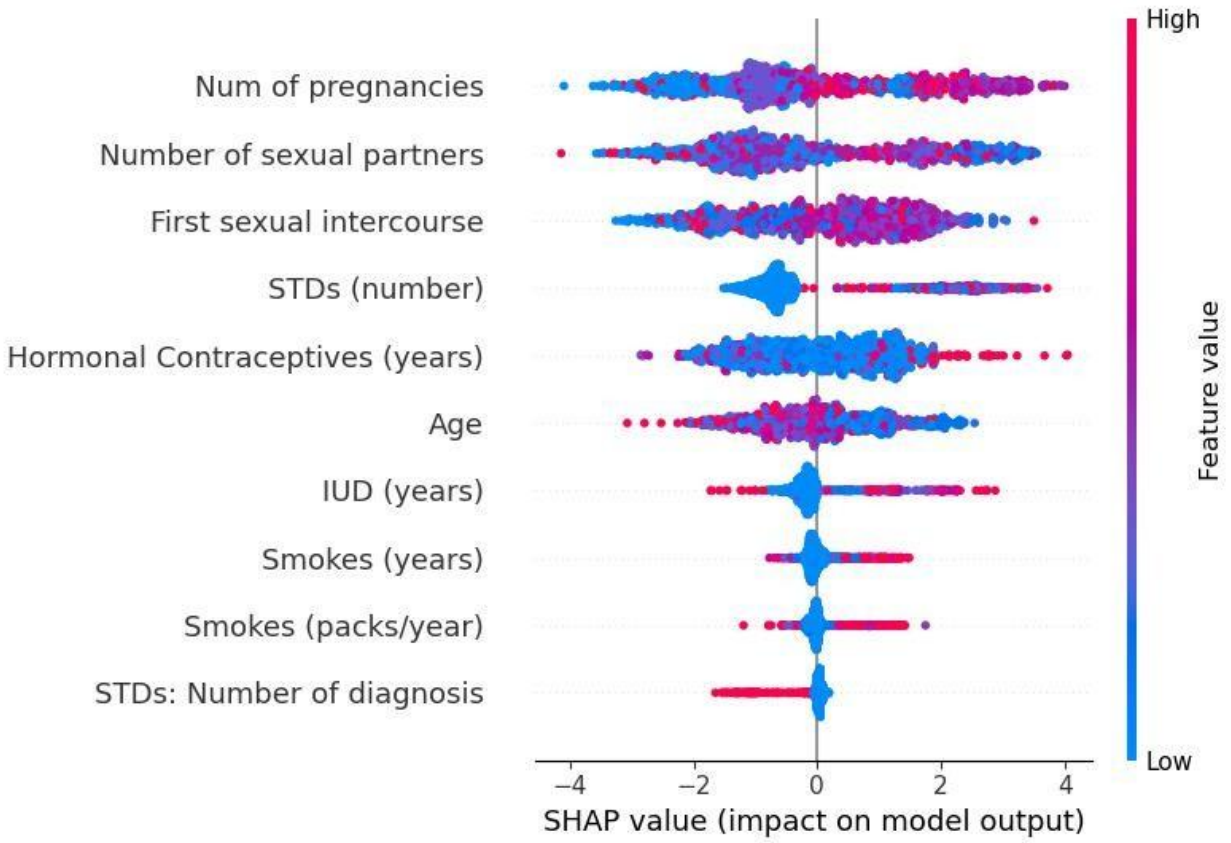
- ***Le choix du modèle CatBoost comme meilleur modèle repose sur plusieurs critères observés lors des tests :***
  - ***Performances élevées : CatBoost a obtenu une précision et un rappel quasi parfaits sur l'ensemble d'entraînement et de test, avec un f1-score élevé, indiquant un bon équilibre entre précision et rappel.***

- ***Robustesse face aux déséquilibres : Bien que nos classes soient initialement déséquilibrées, CatBoost a su bien généraliser après l'application de SMOTE et la normalisation des données.***
- ***Interprétabilité : L'analyse des features importance et des valeurs SHAP montre que CatBoost permet une meilleure compréhension des variables influentes dans la prédiction.***

➢ ***Meilleur compromis entre biais et variance : Contrairement à d'autres modèles comme***

- ***XGBoost ou Random Forest, qui montrent parfois des signes de surdapprentissage (overfitting), CatBoost maintient un bon équilibre entre les performances sur l'entraînement et le test.***

➢ ***Nous avons utilisé SHAP (SHapley Additive exPlanations) pour expliquer le fonctionnement du modèle CatBoost sélectionné. Il permet de visualiser l'impact de chaque variable sur les prédictions du modèle en calculant les valeurs SHAP pour l'ensemble d'entraînement. Le summary plot affiche l'importance des caractéristiques et leur influence sur les décisions du modèle, facilitant ainsi l'interprétation des résultats.***



visualisation de features ayant plus d,impacts sur le cancer

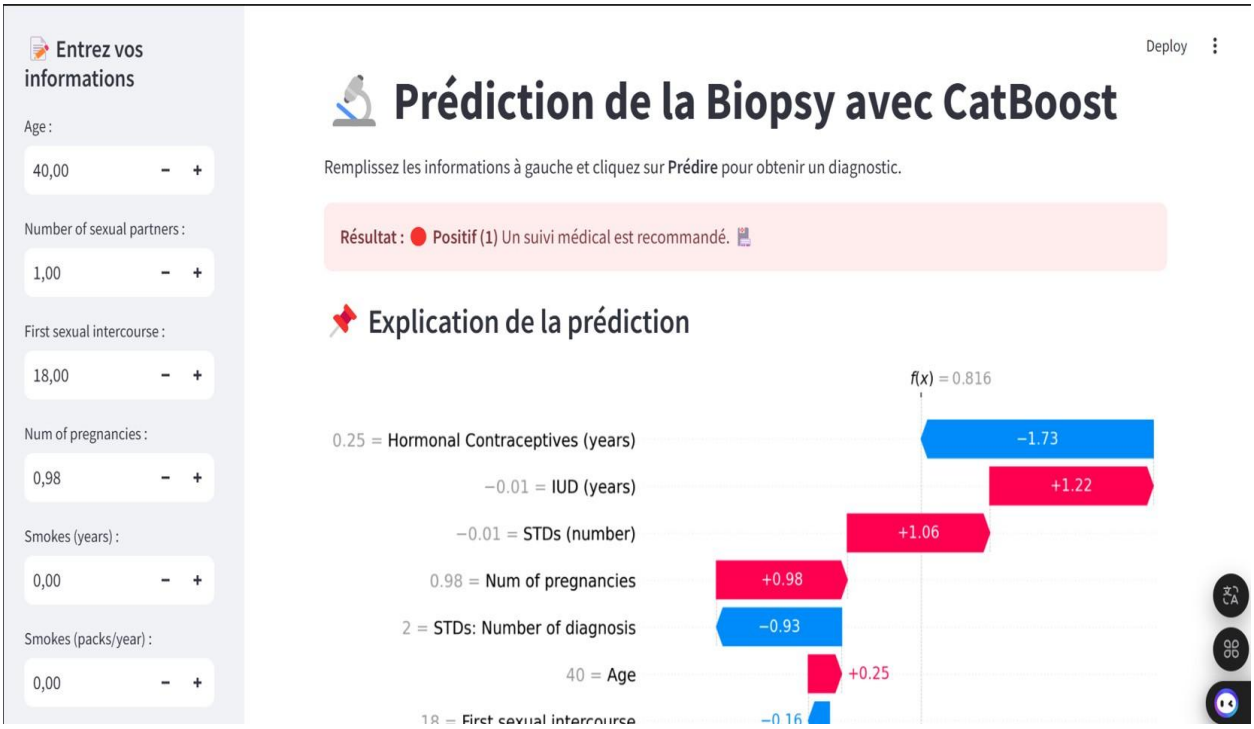
L'application Streamlit développée permet une interaction simple et intuitive pour la prédiction de la biopsie à l'aide du modèle CatBoost. L'interface soignée offre une expérience utilisateur fluide, avec une saisie des données facilitée via la barre latérale.

Une fois les informations renseignées, l'utilisateur peut obtenir un diagnostic immédiat, présenté de manière claire avec un code couleur (vert pour négatif, rouge pour positif). L'ajout de l'explication SHAP améliore la transparence du modèle, en affichant l'impact des différentes caractéristiques sur la prédiction.

Enfin, l'intégration de visuels et d'améliorations esthétiques via du CSS personnalisé rend l'application agréable et accessible, tout en maintenant une rigueur scientifique adaptée à une utilisation médicale.

Pour vérifier que notre application fonctionne bien nous avons pris une ligne dans notre data ou la biopsy donne 1 pour tester.

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives (years)	IUD (years)	STDs (number)	STDs: Number of diagnosis	Biopsy
6	51	3.0	17.0	6.0	34.000000	3.400000	0.00	7.0	0.0	0	1
22	40	1.0	18.0	1.0	0.000000	0.000000	0.25	0.0	2.0	1	1
23	40	1.0	20.0	2.0	0.000000	0.000000	15.00	0.0	0.0	0	1



Capture d,ecran d,une expérimentation de notre interface

Nous obtenons effectivement 1 a la sortie avec la l'explication prédictive avec shap

Selon shap :

Les variables ayant le plus d'impact sont "Num of pregnancies", "Number of sexual partners", "First sexual intercourse" et "STDs (number)".

Une valeur SHAP positive signifie une influence en faveur d'un diagnostic positif (cancer probable).

Une valeur SHAP négative signifie une influence en faveur d'un diagnostic négatif.

La couleur indique la valeur de la caractéristique : bleu (valeurs faibles) et rose (valeurs élevées).

VI. Optimisation de l'Interprétabilité et de l'Expérience Utilisateur grâce à l'Ingénierie des Invites

L'ingénierie des invites (prompt engineering) a joué un rôle clé dans l'amélioration de l'interprétabilité et de la performance de notre application de prédiction de biopsie basée sur CatBoost.

1. Meilleure Explication du Modèle :

- ***Des invites précises ont permis de générer des visualisations SHAP plus claires, facilitant la compréhension des décisions du modèle.***
- ***Une structuration soignée des invites a aidé à mettre en évidence les facteurs les plus influents dans la prédiction des biopsies.***

2. Expérience Utilisateur Optimisée :

- ***Des invites bien conçues dans Streamlit ont guidé les utilisateurs dans la saisie de valeurs pertinentes, réduisant les erreurs d'entrée.***
- ***Les résultats de prédiction (" Négatif (0)" ou " Positif (1)") ont été formulés de manière intuitive et facilement interprétable.***

3. Amélioration de la Performance & du Débogage :

- ***L'expérimentation avec différentes formulations d'invites a permis d'obtenir des explications SHAP concises et pertinentes.***
- ***L'ingénierie des invites a également aidé à valider le choix de CatBoost comme meilleur modèle en mettant en avant les variables les plus influentes.***

VII. Optimisation de la mémoire

L'optimisation de la mémoire n'a pas été une priorité dans notre projet, principalement parce que nous avons utilisé des ordinateurs dotés d'une architecture 64 bits. Cette architecture permet de gérer efficacement de grandes quantités de mémoire, avec une capacité d'adressage bien supérieure à celle des systèmes 32 bits.

En effet, les systèmes 64 bits peuvent théoriquement adresser jusqu'à 16 exaoctets de mémoire, bien que les limites pratiques soient souvent bien en deçà en fonction du matériel et du système d'exploitation. Grâce à cette capacité, la gestion des ensembles de données volumineux et des calculs intensifs est facilitée, réduisant ainsi le besoin d'optimisation mémoire agressive.

De plus, notre implémentation avec CatBoost et SHAP repose sur des algorithmes bien optimisés qui tirent parti des ressources matérielles modernes, minimisant ainsi l'impact d'une éventuelle surcharge mémoire. Ainsi, l'efficacité de notre infrastructure matérielle a rendu inutile une optimisation spécifique de la consommation mémoire.

Mais ci-dessous le code qui nous permettrait de faire l'optimisation de la mémoire

```python
def optimize_memory(df):
    for col in df.columns:
        if df[col].dtype == 'float64':
            df[col] = df[col].astype('float32')
        elif df[col].dtype == 'int64':
            df[col] = df[col].astype('int32')
    return df

df_cleaned = optimize_memory(df_cleaned)
print("Mémoire optimisée.")
```

Le code pour optimiser la mémoire

VIII.Conlusion

Ce projet nous a permis d'explorer l'utilisation de l'intelligence artificielle, et plus particulièrement du modèle CatBoost, pour la prédiction du cancer du col de l'utérus. À

travers l'analyse des données et l'interprétation des résultats avec SHAP, nous avons identifié les facteurs influençant le plus les prédictions. Bien que nous soyons débutants, cette première approche nous a permis de mettre en place une application fonctionnelle qui pourrait servir de base pour des développements futurs et une amélioration continue.

Table des figures

https://fr.planet-health.be/sante-feminine/des-techniques-de-detection-innovantes-du-cancer-du-col-de-luterus-2/

https://www.sciencesetavenir.fr/assets/img/2017/03/24/cover-r4x3w1200-5c3863545eda0-papillomavirus.jpg

https://inseefrlab.github.io/DT_methodes_ensemblistes/chapters/chapter2/4-boosting.html

https://medium.com/@priyankaparashar54/random-forest-classification-and-its-mathematical-implementation-f5e7e6878ca