

Rapport

chef d'oeuvre

Le cancer du sein et l'intelligence artificielle

Titre professionnel «**Développeur en intelligence artificielle**»
de niveau 6 enregistré au RNCP sous le n°34757
Passage par la voie de la formation - parcours de 19 mois achevé en février 2023

PROST VERONIQUE

TABLE DES MATIÈRES

INTRODUCTION

Le contexte
Le projet et objectifs
Environnement de travail
Synthèse des données

1. ANALYSE EXPLORATOIRE DES DONNÉES

1.1 Valeurs manquantes, nulles, dupliquées
1.2 Valeurs aberrantes
1.3 Distribution des données
1.4 Corrélation
1.5 La cible

2. PRÉ-TRAITEMENT DES DONNÉES

2.1 Des données déséquilibrées
2.2 Choix des variables
2.3 Normalisation
2.4 Fractionnement

3. MODÈLE D'INTELLIGENCE ARTIFICIELLE

3.1 Choix de l'algorithme
3.2 Analyse des résultats
3.3 Importance des variables

4. DÉVELOPPEMENT DE L'APPLICATION

4.1 Front-end et back-end
4.2 Base de données MySQL
4.3 Tests unitaires, fonctionnels et de non régression
4.4 Déploiement sur Microsoft Azure
4.5 Monitoring et alerte

LE CONTEXTE



Le cancer du sein est le cancer le plus fréquent chez les femmes dans le monde. Il représente 25 % de tous les cas de cancer et a touché plus de 2,2 millions de personnes en 2020.

MÉTHODES DE DIAGNOSTIC

Le diagnostic est une étape importante du traitement du cancer du sein qui influence lourdement le pronostic de la patiente. Pris en charge dès ses prémices, un cancer du sein offre en effet davantage de chance de survie.

De fait, la science s'applique depuis longtemps à élaborer des méthodes de dépistage et de diagnostic permettant de mieux identifier et catégoriser les tumeurs mammaires.

LE DÉPISTAGE

Le dépistage repose sur une mammographie (radiographie des seins), associée à un examen clinique des seins (observation et palpation).

En cas d'anomalie indéterminée ou suspecte, il est proposé au patient de pratiquer une biopsie par aiguille fine (BAF) afin de prélever des cellules dans la masse. Le prélèvement est ensuite examiné au microscope (*voir image ci-jointe*).

Les caractéristiques des cellules sont ainsi étudiées, de part leur texture, périmètre, symétrie, ect... confirmant ou non la présence de tumeur maligne ou bénigne (exempt de métastase, non cancéreuse) diagnostiqué par le professionnel de santé et/ou par l'intelligence artificielle.

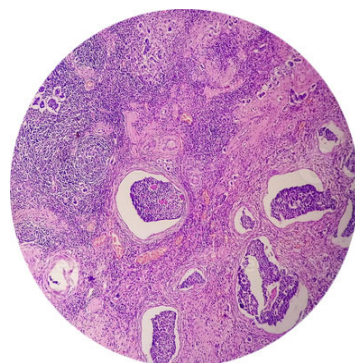


Image microscopique de cellules touchées par le cancer du sein

LE CANCER DU SEIN ET LA SCIENCE

Les données recueillies sur les cellules et l'intelligence artificielle offrent ainsi une aide convaincante aux professionnels de santé en palliant aux erreurs humaines et en réduisant considérablement le temps d'analyse.

Par exemple, avec 99,3% des cancers détectés, le modèle de deep learning du logiciel LYNA, dépasse de loin les performances des experts médicaux qui ne parviennent à identifier les petites métastases que dans 62.8% des cas.

**Caractéristiques
d'une cellule**
(unité : micromètre)

rayon

texture

région

compacité

concavité

symétrie

dimension fractale

périmètre

lissé



LE PROJET &

OBJECTIFS

Dans l'objectif d'une aide au diagnostic, et grâce à un modèle d'apprentissage supervisé, nous allons classifier des cas de tumeurs malignes ou bénignes grâce aux différentes caractéristiques des noyaux cellulaires.

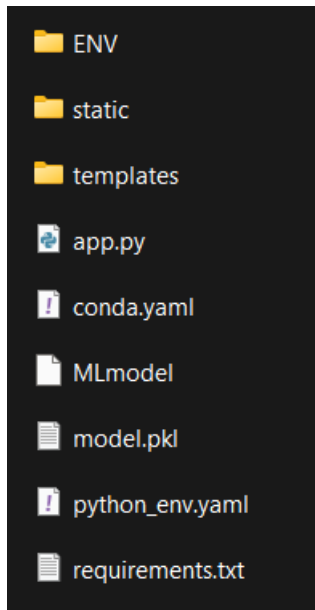
Notre modèle répondra aux besoins des spécialistes de l'image médicale et oncologue.

Une application sera mise à disposition des professionnels de santé afin d'affirmer ou non la présence d'une tumeur cancéreuse.

ENVIRONNEMENT DE TRAVAIL

1.1 Environnement

Un environnement virtuel de développement en Python a été créé grâce à la librairie **virtualenv**. Cet environnement, isolé du système, est indispensable au bon développement d'une application.



Le dossier du projet contient :

- Dossier *ENV* : contient l'environnement Python
- Dossiers *static* et *templates* : contient les fichiers nécessaires à l'application développée avec la librairie **Flask**
- Fichier *app.py* : contient des API Flask qui reçoivent l'entrée de données, calcule la valeur de notre prédiction et l'affiche
- Fichiers *conda.yaml*, *MLmodel*, *python_env.yaml* : fichiers dépendants de notre modèle enregistré dans **MLFlow**
- Fichier *model.pkl* : modèle pré-entraîné de machine learning
- Fichier *requirements.txt* : fichier contenant toutes les librairies dépendantes de notre application

1.2 Stockage des données

Les données, au format .csv, sont stockées dans une base de données **MySQL**. Les données pourront être exploitées pleinement (lecture, transformation, écriture) tout au cours de l'exploration, développement du modèle et application.

1 • `SELECT * FROM data.breast_cancer;`

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
0	M		17.99	10.38	122.8	1001	0.1184	0.2776
1	M		20.57	17.77	132.9	1326	0.08474	0.07864
2	M		19.69	21.25	130	1203	0.1096	0.1599
3	M		11.42	20.38	77.58	386.1	0.1425	0.2839
4	M		20.29	14.34	135.1	1297	0.1003	0.1328

Capture d'écran MySQL Workbench

1.3 Développement

Le projet a été développé avec les IDE (environnement de développement intégré) **Jupyter Notebook** et **Visual Studio Code**.

1.4 Sauvegarde du projet

La totalité du projet est sauvegardé et disponible sur **GitHub** :
https://github.com/bonjourcerise/dev_ia

SYNTHÈSE DES DONNÉES



SOURCE DES DONNÉES

Le jeu de données provient de l'Université du Wisconsin.

Il décrit les caractéristiques de noyaux cellulaires, en micromètres, de masses mammaires issues d'images microscopiques de prélèvement par BAF (biopsie à l'aiguille fine), sur un échantillon de plus de 500 patients.



CRÉATEURS

Dr. William H. Wolberg, Département de chirurgie générale, Université du Wisconsin

W. Nick Street et Olvi L. Mangasarian, Département Computer Sciences, Université du Wisconsin



UTILISATIONS

Ce jeu de données a été étudié et cité dans le cadre de littérature médicale.

W.N. Street, W.H. Wolberg, O.L. Mangasarian

Nuclear feature extraction for breast tumor diagnosis

S&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993

O.L. Mangasarian, W.N. Street, W.H. Wolberg

Breast cancer diagnosis and prognosis via linear programming

Operations Research, 43(4), pages 570-577, July-August 1995.

W.H. Wolberg, W.N. Street, O.L. Mangasarian

Machine learning techniques to diagnose breast cancer from fine-needle aspirates

Cancer Letters 77 (1994) 163-171



SPÉCIFICATIONS

31 colonnes - 569 lignes

30 colonnes numériques - 1 colonne catégorielle

Unité des valeurs décrivant les cellules : micromètre (µm)

Aucune valeur manquante, nulle ou dupliquée

Fichier format .csv



TRANSFORMATION

Encodage de la cible : B = 0 / M = 1

Normalisation des données

Sur-échantillonnage de la classe minoritaire

La dernière actualisation du jeu de données date de l'année 1995.

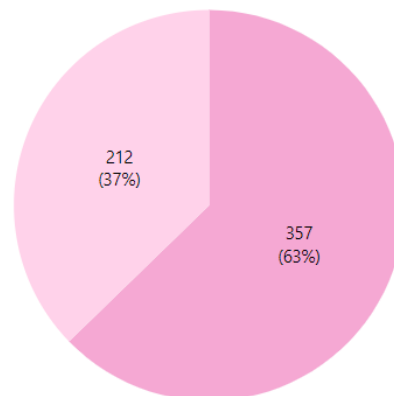


CIBLE

La cible est la colonne "diagnosis"

● B = Tumeur **B**énigne

● M = Tumeur **M**aligne



Répartition de la colonne «diagnosis»

LES DONNÉES ET LE MODÈLE D'IA

1. ANALYSE EXPLORATOIRE DES DONNÉES

L'analyse exploratoire des données est essentielle afin d'avoir une meilleure compréhension des variables d'un ensemble de données, connaître les relations qui existent entre elles, constater et révéler des anomalies, vérifier des hypothèses et enfin évaluer au mieux le nettoyage et pré-traitement à mettre en place pour obtenir des données prêtes à être entraînées. La visualisation des données s'inscrit dans cette analyse par représentation graphique des informations.

Les données collectées sur le site de l'Université du Wisconsin sont au format CSV (*Comma Separated Values*). J'ai effectué l'analyse avec les bibliothèques Python suivantes : **Pandas, Numpy, Seaborn, Matplotlib**.

Le jeu de données compte 10 variables décrivant les caractères d'un noyau cellulaire en micromètres avec pour chaque variable, une caractéristique supplémentaire : la moyenne (*mean*), l'erreur type (*standard error*) et la plus grande (*worst*) soit un total de 30 variables quantitatives de type *float* (nombre décimal) et 1 variable booléenne, notre cible, que j'ai encodé en numérique, indispensable pour le processus de modélisation.

Nom des variables	Unité	Description
diagnosis	Classe 0 - 1	Notre cible 0 = Tumeur bénigne / 1 = Tumeur Maligne
radius	Micromètre	Rayon
texture	Micromètre	Texture
périmètre	Micromètre	Périmètre
area	Micromètre	Région
smoothness	Micromètre	Lisse
compactness	Micromètre	Compacité
concavity	Micromètre	Concavité
concave points	Micromètre	Points concaves
symmetry	Micromètre	Symétrie
fractal dimension	Micromètre	Dimension fractale

1.1 Valeurs manquantes, nulles, dupliquées

Les fonctions *isnull()*, *isna()* et *duplicate()* nous apprennent que nos données ne possèdent aucune valeur manquante, nulle ou dupliquée.

```
-----Missing value-----
diagnosis          0
radius_mean        0
texture_mean       0
perimeter_mean     0
area_mean          0
smoothness_mean    0
compactness_mean   0
concavity_mean     0
concave points_mean 0
```

```
-----Null value-----
diagnosis          0
radius_mean        0
texture_mean       0
perimeter_mean     0
area_mean          0
smoothness_mean    0
compactness_mean   0
concavity_mean     0
concave points_mean 0
```

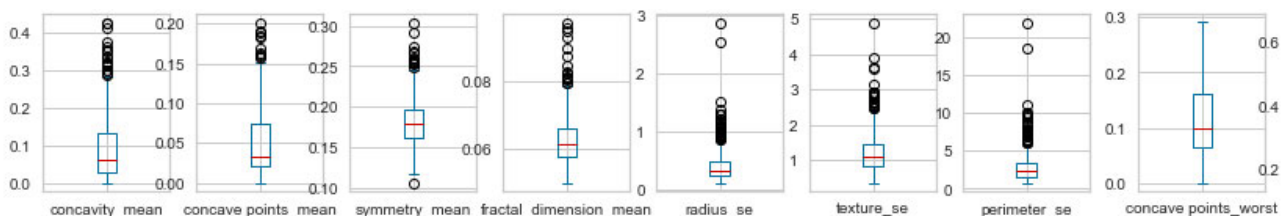
Réponse à la requête Python cherchant les valeurs manquantes et nulles

LES DONNÉES ET LE MODÈLE D'IA

1.2 Valeurs aberrantes ou atypiques ?

La fonction `describe()` décrit les valeurs minimum, maximum, moyenne, d'écart type de chaque variable et nous permet d'observer les valeurs aberrantes (*outliers*), des valeurs extrêmes, anormalement faibles ou élevées de la tendance globale des autres observations. Leur présence dans les données peut conduire à des estimateurs de paramètres biaisés et à une interprétation erronée des résultats.

Grâce à la visualisation, je décèle de nombreuses valeurs aberrantes, comme on le constate ci-dessous sur les *boxplots* : chaque point est une valeur extrême. Ce constat s'avère vérifiable sur toutes nos variables sauf *concave points_worst*.



Une fonction pour détecter les valeurs supérieures au troisième quartile et les valeurs inférieures au premier quartile avec la méthode **IQR** (InterQuartile Range) confirme la visualisation.

```
1 outlier(s) in radius_mean
1 outlier(s) in texture_mean
5 outlier(s) in area_mean
1 outlier(s) in smoothness_mean
1 outlier(s) in compactness_mean
1 outlier(s) in symmetry_mean
4 outlier(s) in fractal_dimension_mean

7 outlier(s) in radius_se
4 outlier(s) in texture_se
12 outlier(s) in perimeter_se
22 outlier(s) in area_se
7 outlier(s) in smoothness_se
6 outlier(s) in compactness_se
6 outlier(s) in concavity_se

3 outlier(s) in concave points_se
9 outlier(s) in symmetry_se
11 outlier(s) in fractal_dimension_se
1 outlier(s) in perimeter_worst
7 outlier(s) in area_worst
3 outlier(s) in compactness_worst
1 outlier(s) in concavity_worst
```

Nombres de valeurs aberrantes dans chaque variable du jeu de données

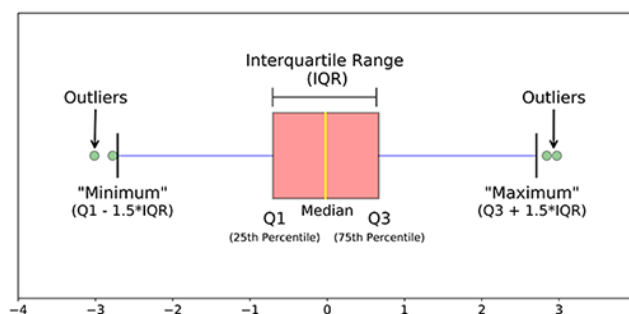


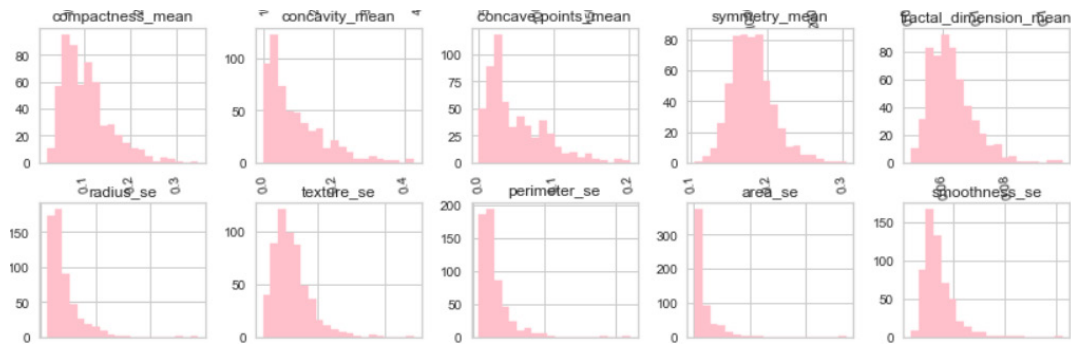
Schéma explicatif de la méthode IQR pour détecter les valeurs aberrantes

Une valeur aberrante n'est pas forcément une valeur erronée, elle peut être une information intéressante, une valeur atypique. En cas de suppression, nous pourrions perdre des informations réelles et je possède déjà un nombre limité de données. Je décide de les conserver dans le modèle de machine learning en accord avec l'expert-métier.

LES DONNÉES ET LE MODÈLE D'IA

1.3 Distribution des données

J'ai observé par représentation graphique grâce à des *histogrammes*, que la majorité de nos 30 variables ne suivent pas une distribution gaussienne - ou en cloche - mais une distribution asymétrique, expliquée par le nombre de valeurs aberrantes supérieures.



1.4 Corrélation des variables

Toujours dans l'optique de créer un modèle fiable, il est important d'identifier les variables dépendantes : la corrélation. En Python, nous pouvons utiliser la fonction `corr()` qui utilise la *formule de Pearson* soit le calcul de la covariance entre les variables, divisée par le produit de leurs écarts types :

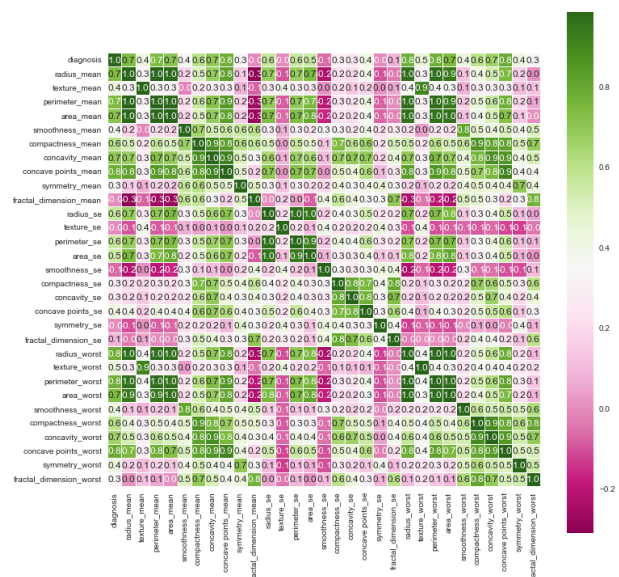
$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

Plus la valeur du coefficient de corrélation est élevée, plus les deux variables sont dépendantes. Le résultat est compris entre -1 et 1.

En liant la fonction `corr()` et une carte thermique (*heatmap*), on obtient une représentation graphique des plus lisible afin de lire les coefficients de corrélations entre les variables du jeu de données.

Sur ce visuel, on constate une forte corrélation, supérieur à 0.95 entre les variables suivantes :

- `radius_mean` et `perimeter_mean`
- `radius_mean` et `area_mean`
- `radius_worst` et `perimeter_worst`
- `perimeter_mean` et `area_worst`
- `compactness_worst` et `concavity_worst`
- `concave points_mean` et `perimeter_mean`
- `area_se` et `perimeter_se`



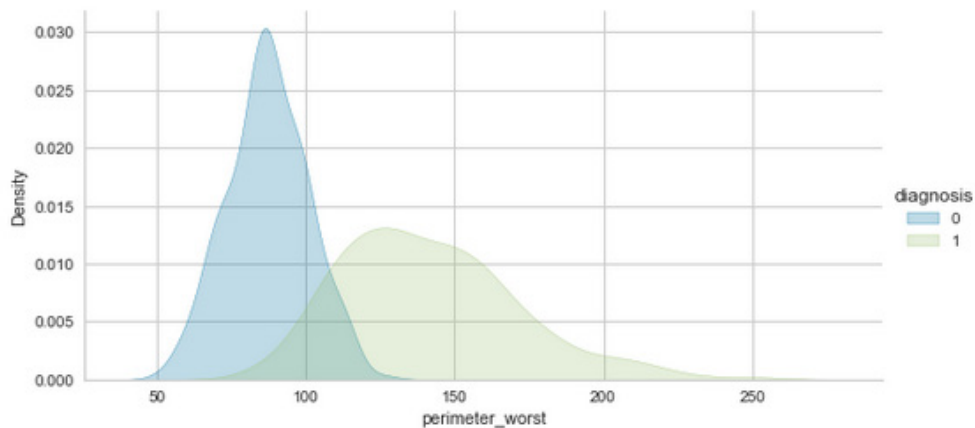
LES DONNÉES ET LE MODÈLE D'IA

1.5 La cible "diagnosis"

La fonction *FacetGrid* de la librairie de visualisation **Seaborn** m'a permis de créer des graphes multiples en fonction des 2 classes de notre cible :

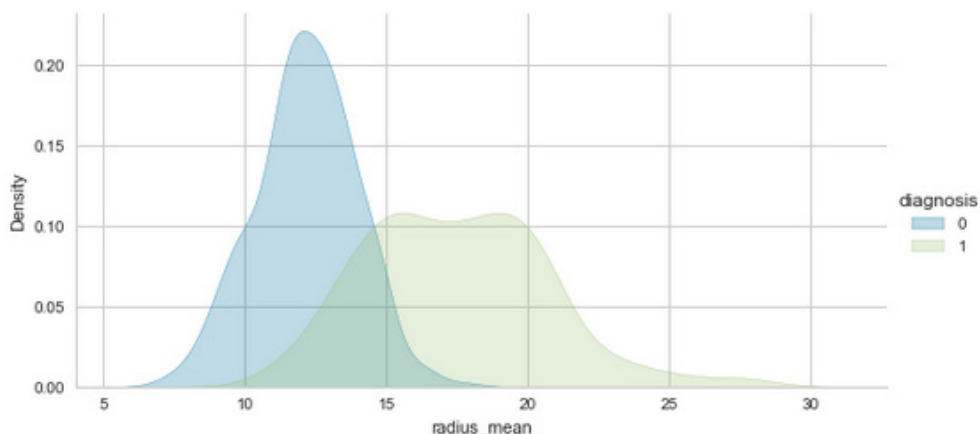
- 0 = tumeur bénigne
- 1 = tumeur maligne

On remarque bien sur le visuel ci-dessous que la distribution de la variable *perimeter_worst* est totalement différente entre la classe 0 et 1. Je peux en conclure que le périmètre d'un noyau cellulaire d'une tumeur bénigne est supérieur à celui d'une tumeur maligne.



Distribution de perimeter_worst en fonction des classes de diagnosis

Ici, on remarque également que la moyenne du rayon d'un noyau cellulaire d'une tumeur bénigne est supérieure par rapport à celle d'une tumeur maligne.



Distribution de radius_mean en fonction des classes de diagnosis

Il existe donc, dans notre jeu de données, des différences évidentes entre les 2 classes.

LES DONNÉES ET LE MODÈLE D'IA

2. PRÉ-TRAITEMENT DES DONNÉES

Les données nécessitent un pré-traitement (*preprocessing*) avant la modélisation.

2.1 Des données déséquilibrées

Nos données d'apprentissage sont déséquilibrées avec 63% de détection de tumeur bénigne et 37% de tumeur maligne. C'est une situation fréquente rencontrée dans nombre de problèmes réels comme dans la détection de fraude ou ici, le diagnostic médical.

Un modèle de classification fonctionne correctement si les proportions des classes sont représentées de façon égale dans l'échantillon, il est donc essentiel d'avoir à disposition des données équilibrées, un bon ajustement afin d'éviter les 2 problèmes en *Machine Learning* :

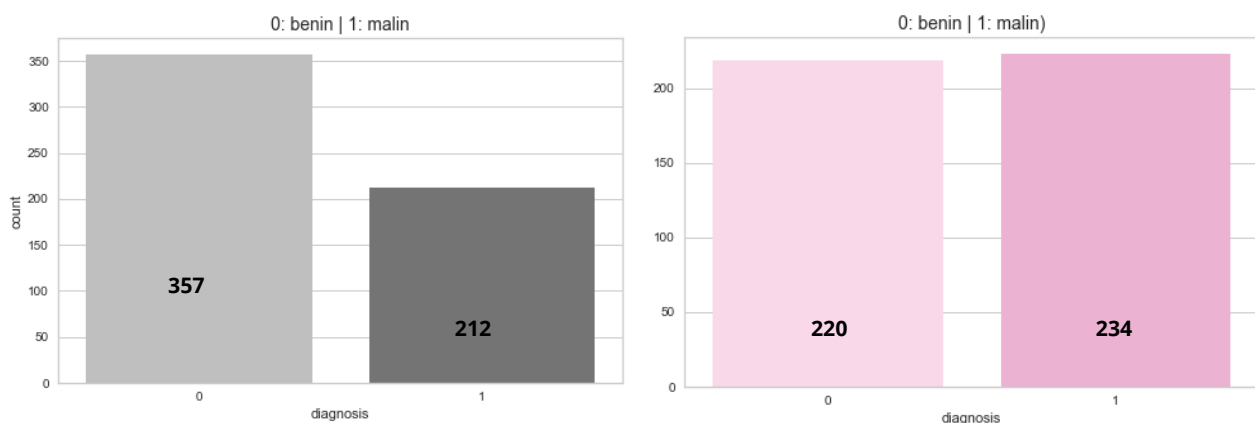
- le risque d'*overfitting* (sur-apprentissage) : le modèle prédictif pourra donner de très bonnes prédictions sur les données d'entraînement (les données qu'il a déjà "vues"), mais il prédira mal sur des données qu'il n'a pas encore vues
- le risque d'*underfitting* (sous-apprentissage) : le modèle est incapable de saisir la tendance des données, il ignore des entités et produit, par conséquent, des prédictions incorrectes

Afin de rééquilibrer notre jeu de données, il existe plusieurs méthodes comme le ré-échantillonnage

- L'*oversampling* (sur-échantillonnage) : augmenter les données de la classe minoritaire
- L'*undersampling* (sous-échantillonnage) : supprimer des données de la classe majoritaire, on perd toutefois des informations précieuses

Dans mon cas, je vais combiner ces 2 méthodes pour se rapprocher d'une situation équilibrée en utilisant la méthode **SMOTEENN** de la librairie Python **imbalanced-learn**. **SMOTEENN** combine l'*oversampling* avec **SMOTE** (Synthetic Minority Oversampling) et l'*undersampling* avec **ENN** (Edited Nearest Neighbor).

Répartition des classes de la colonne *Diagnosis* avant et après utilisation de **SMOTEENN**



LES DONNÉES ET LE MODÈLE D'IA

2.2 Choix des features

Le choix des variables a gardé dans mon modèle a été réalisé avec l'aide de l'expert-métier.

Je conserve uniquement les 11 entités suivantes : *concave points_worst*, *concave points_mean*, *radius_worst*, *perimeter_worst*, *compactness_worst*, *symmetry_worst*, *texture_worst*, *area_se*, *concavity_mean*, *area_worst*, *texture_mean*

2.3 Normalisation des données

J'utilise la fonction *MinMax* de la librairie **Scikit-learn** pour normaliser les données. La distribution des valeurs va être redimensionnée de sorte à avoir des données normalement distribuées entre 0 et 1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Formule de la fonction *MinMaxScaler*

2.4 Fractionnement des données

J'utilise la fonction *train_test_split* de la librairie **Scikit-learn** afin de fractionner aléatoirement le jeu de données en 2 : une partie servira à l'entraînement du modèle, l'autre partie servira au test. Je consacre 70% des données à l'entraînement et 30% au test.

```
# split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)
```

Fonction *train_test_split* de *Scikit-learn*

Le jeu de données est désormais prêt à être modélisé.

LES DONNÉES ET LE MODÈLE D'IA

3. MODÈLE DE MACHINE LEARNING

Mon modèle de *Machine Learning* sera de l'*apprentissage supervisé*, toutes les données sont étiquetées, de type *classification* sur la cible, la variable *diagnosis*.

3.1 Choix de l'algorithme

Il existe différentes librairies *Python* pour entraîner un modèle de classification : **Scikit-Learn**, **PyTorch** ou encore **Tensorflow**.

J'ai testé des algorithmes de **Scikit-Learn**, la librairie de référence pour faire du *Machine Learning* en *Python* afin d'obtenir le meilleur modèle possible : le résultat d'une classification comprenant le moins de *faux positifs* et *faux négatifs*.

En effet, le résultat immédiat et lisible d'un algorithme de classification est représenté par une *matrice de confusion* contenant les données classifiées de notre modèle comme suit : *faux positif*, *faux négatif*, *vrai positif*, *vrai négatif*. Les *faux* représentent les valeurs où le modèle s'est trompé.

		Classe réelle	
		0 Tumeur bénigne	1 Tumeur maligne
Prédiction	0 Tumeur bénigne	Vrai négatif	Faux négatif
	1 Tumeur maligne	Faux positif	Vrai positif

Matrice de confusion

Chaque itération a été sauvegardée dans **MLflow**, un outil open source qui permet de gérer le cycle de vie des modèles de *Machine Learning*.

Les modèles ainsi que les hyper-paramètres associés sont stockés, monitorés et peuvent être réutilisés de manière efficace. Parmi les principaux avantages de **MLflow**, on note le suivi des expériences pour comparer les paramètres et résultats, la gestion et déploiement de modèles de diverses librairies de *Machine Learning*, la création de package pour chaque modèle afin de les partager ou passer en production et même l'hébergement de modèles **MLflow** en tant que points de terminaison *API REST*.

LES DONNÉES ET LE MODÈLE D'IA

Voici un aperçu de tous les modèles testés ainsi que leurs métriques :

mlflow1.26.1

ExperimentsModels

GitHubDocs

Experiments

CL breast cancer

Share

Search Experiments

Experiment ID: 1

DescriptionEdit

Refresh

Compare

Delete

Download CSV

↓ AUC

All time

Columns

Only show differences

metrics.rmse < 1 and params.model = "tree"

Search

Filter

Clear

Showing 8 matching runs

	Start Time	Durati	Run Name	Us	Sc	Ve	Models	↓ AUC	Accuracy	F1-Score	Precision	Recall
	17 hours ago	5.1s	Light GBM CL				sklearn	0.969	0.972	0.96	0.964	0.957
	17 hours ago	5.7s	ACP Logistic Regression				sklearn	0.953	0.958	0.947	0.971	0.924
	17 hours ago	4.8s	Logistic Regression				sklearn	0.95	0.958	0.943	0.969	0.918
	17 hours ago	5.2s	SMOTEENN Logistic Regression				sklearn	0.947	0.953	0.938	0.952	0.923
	17 hours ago	6.0s	FS Logistic Regression				sklearn	0.942	0.943	0.924	0.908	0.94
	17 hours ago	4.8s	Decision Tree CL				sklearn	0.897	0.9	0.868	0.852	0.886
	8 seconds ago	4.8s	SMT Voting Classifier				sklearn	-	0.952	0.93	0.955	0.907
	17 hours ago	5.3s	SMT GS Random Forest				sklearn	-	0.959	0.952	0.921	0.986

L'algorithme de classification **Random Forest Classifier** a été retenu avec le résultat le plus satisfaisant, notamment sur la réduction des *faux positifs* et *faux négatifs*. Sur 171 patients, il y a 4 diagnostics erronés soit moins de 2,5% d'erreurs.

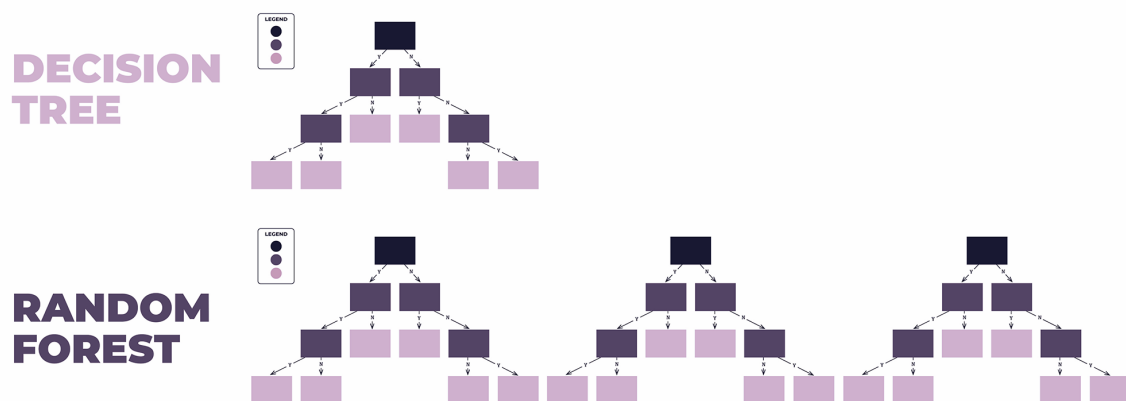


Matrice de confusion du modèle Random Forest Classifier

LES DONNÉES ET LE MODÈLE D'IA

L'algorithme **Random Forest Classifier**, méthode d'ensemble et de *bagging* introduite par Leo Breiman et Adele Cutler en 2001, est appelé *forêt* parce qu'il génère la croissance d'une forêt d'arbres de décision (*decision trees*).

Les données de ces *arbres* sont ensuite fusionnées pour garantir les prédictions les plus précises. Alors qu'un arbre de décision isolé a un seul résultat et une gamme étroite de groupes, la *forêt* assure un résultat plus précis en comptant sur un plus grand nombre de groupes et de décisions.



Pour obtenir ce résultat avec l'algorithme **Random Forest Classifier**, j'ai utilisé une méthode d'optimisation du modèle, **GridSearchCV**.

Cet outil permet d'obtenir un modèle plus robuste en testant toute une série de paramètres d'un algorithme et de comparer leur performance pour en déduire les meilleures paramètres.

Dans notre cas, j'ai demandé à **GridSearchCV** de créer un modèle **Random Forest Classifier** pour chaque combinaison de ces paramètres :

- Nombre d'arbres (*n_estimators*) : 50, 100, 150
- Profondeur maximale de l'arbre (*max_depth*) : 2, 6, 8, 10

```
param_grid = {'n_estimators': [50, 100, 150, 200, 400],  
              'max_depth': [2, 4, 6, 8, 10]  
            }
```

Code Python

GridSearchCV obtient donc 9 modèles à construire et pour valider la fiabilité de ces modèles, il va utiliser une méthode de **validation croisée** (*K-Fold Cross Validation*) : l'assurance d'évaluer les modèles sur des échantillons jamais vus.

En effet, il est très important de tester la stabilité de son modèle de *Machine Learning* en évaluant sa performance avec des données encore inédites. En se basant sur les résultats de ce test, on pourra juger si un *sur-apprentissage* ou un *sous-apprentissage* a eu lieu.





- Les paramètres du meilleur modèle : `{ 'max_depth': 8, 'n_estimators': 50 }`

LES DONNÉES ET LE MODÈLE D'IA

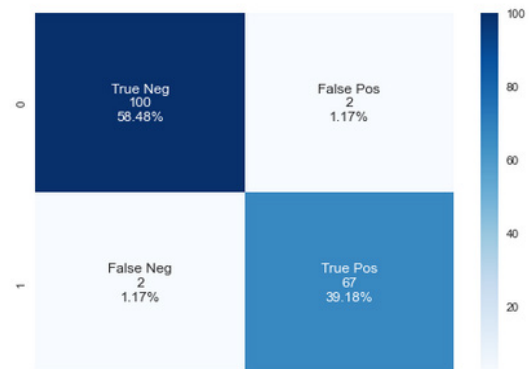
3.2 Analyse des résultats

Comme vu dans le chapitre précédent, le modèle a obtenu moins de 2,5% d'erreurs et pour l'évaluer il existe plusieurs métriques possédant un seuil de 0 à 1. Dans le cas d'un apprentissage supervisé de classification, on utilise les métriques suivantes, basées sur la matrice de confusion: l'*accuracy*, le *recall*, la *precision* et le *F1-score*.

SCORE DE MON MODELE

Name	Value
Accuracy 	0.959
F1-Score 	0.952
Precision 	0.921
Recall 	0.986

MATRICE DE CONFUSION



L'accuracy

L'*accuracy* permet de décrire la performance du modèle sur les individus positifs et négatifs, de façon symétrique, avec une valeur unique. Elle mesure le taux de prédictions correctes sur l'ensemble des individus.

$$\text{Accuracy} = \frac{\text{prédictions correctes}}{\text{total des cas}} * 100\%$$

Le F1-score

Le *F1 Score*, que l'on appelle aussi la *moyenne harmonique*, permet d'effectuer une bonne évaluation de la performance d'un modèle en combinant le *recall* et la *precision*.

$$\text{F1-Score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Le recall

Le *recall* mesure la capacité du modèle à détecter l'ensemble des individus positifs. Il correspond au taux d'individus positifs détectés par le modèle.

Parmi les patientes prédites avec une tumeur maligne combien ont été correctement prédites ?

$$\text{Recall} = \frac{\text{vrai positif}}{\text{vrai positif} + \text{faux négatif}}$$

La precision

La *precision* mesure la capacité du modèle à reconnaître le nombre de prédictions positives bien effectuées.

Parmi les patientes prédites avec une tumeur maligne combien le sont réellement ?

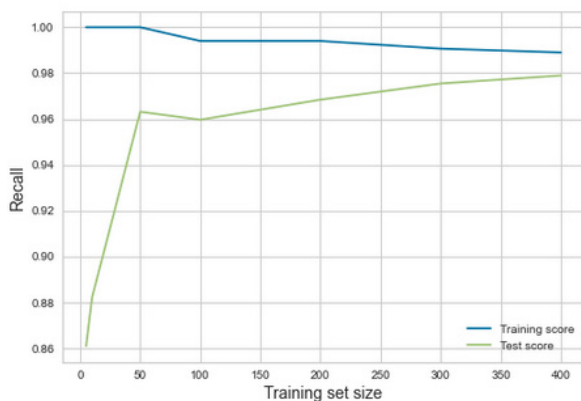
$$\text{Precision} = \frac{\text{vrai positif}}{\text{vrai positif} + \text{faux positif}}$$

La *precision* et le *recall* se concentrent sur la performance du modèle uniquement sur les individus positifs.

LES DONNÉES ET LE MODÈLE D'IA

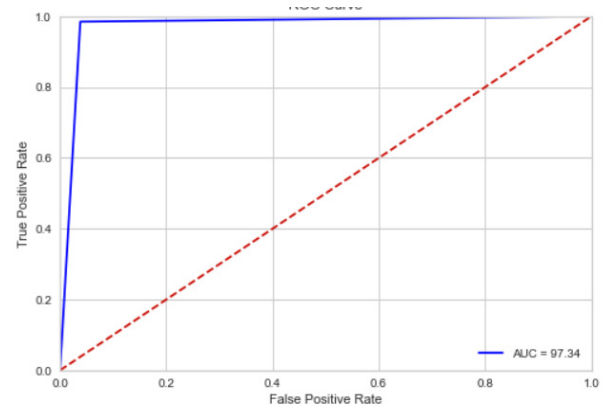
Nous avons aussi à notre disposition, d'autres métriques et visuels pour évaluer la performance et précision de notre modèle.

La courbe d'apprentissage



La *courbe d'apprentissage* est un excellent outil de diagnostic pour déterminer le biais et la variance dans un algorithme d'apprentissage automatique supervisé. Elle nous montre le score de validation et d'apprentissage pour un nombre variable d'échantillons.

La courbe AUC-ROC



La *courbe ROC* (Receiver Characteristic Operator) montre le rapport entre le taux de vrais positifs du modèle et le taux de faux positifs. L'*AUC* (Area Under the Curve) est la mesure de la capacité d'un classificateur à faire la distinction entre les classes.

On peut ainsi conclure que mon modèle :

- avec une **accuracy de 0,96**, a un taux de prédictions correctes sur l'ensemble des individus
- avec un **F1-score de 0,95**, a une très bonne performance
- avec un **recall de 0,98**, a une très bonne capacité à détecter l'ensemble des individus positifs, néanmoins cela ne donne aucune information sur sa qualité de prédiction sur les négatifs
- avec une **precision de 0,92**, a bien prédit que la majorité des individus positifs du modèle sont des positifs

Egalement, l'**AUC de 97,3** (sur 100) nous indique une précision performante du modèle pour distinguer les classes positives et négatives. Quant à la **courbe ROC**, élevée dans le coin supérieur gauche, elle nous indique qu'il est un excellent classificateur, et enfin la **courbe d'apprentissage** démontre son équilibre entre biais et variance.

LES DONNÉES ET LE MODÈLE D'IA

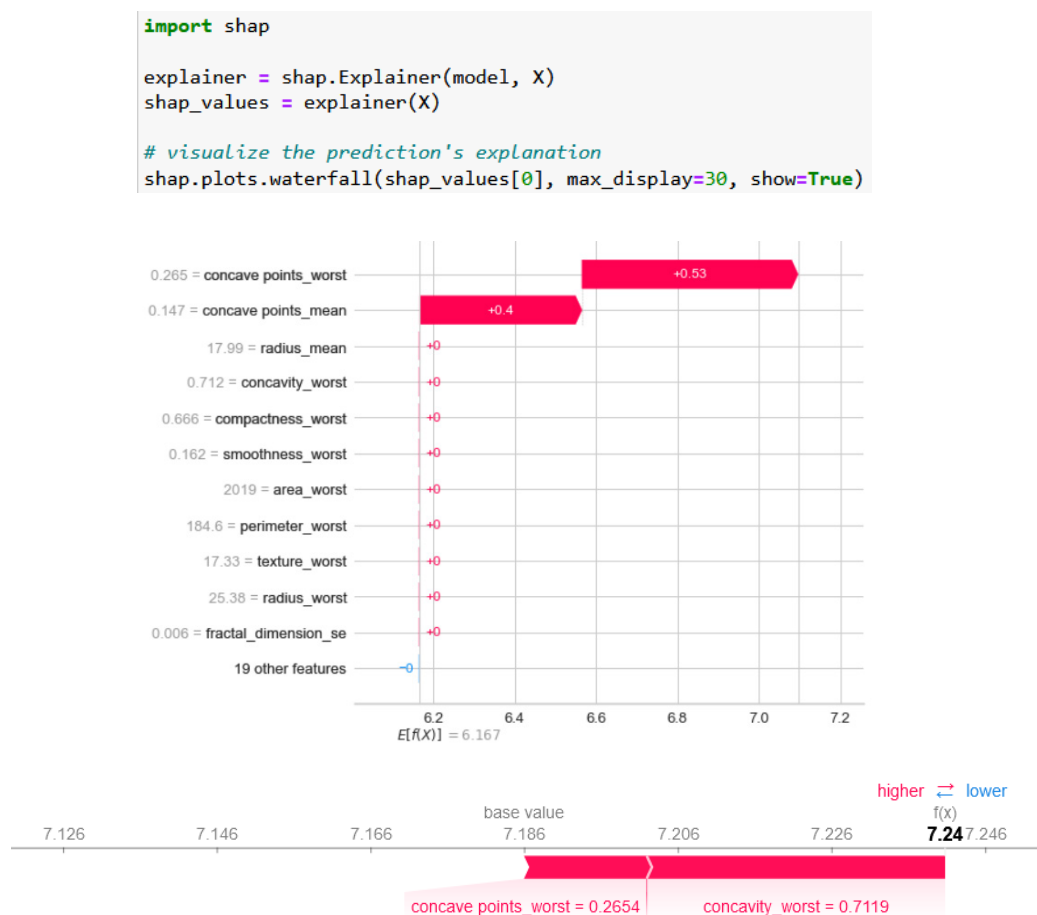
3.3 L'importance des variables dans le modèle

L'importance des variables ou plus communément en anglais, *Features Importance*, est une technique qui calcule un score pour chaque variable d'entrée, dans un modèle. Le score représente ainsi l'importance de la variable pour la prédiction.

L'importance des données est extrêmement utile pour :

- La compréhension des données : cette technique permet de comprendre la relation entre les variables et la cible, ainsi que de détecter les variables non pertinentes pour le modèle.
- L'amélioration du modèle : on peut ainsi simplifier le modèle et optimiser ses performances, en supprimant les variables au score le plus faible.

J'ai utilisé la librairie **SHAP** (*SHapley Additive exPlanations*) afin d'examiner l'importance des variables de mon modèle. **SHAP** a été créé par le laboratoire Su-In-Lee de l'Université de Washington dont l'objectif fondamental est de faire progresser l'intégration de l'IA et ML dans les sciences biomédicales, accompagné de Microsoft Research.

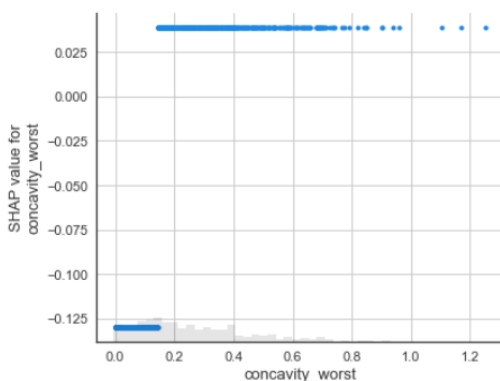


Code Python et visualisation avec librairie SHAP

LES DONNÉES ET LE MODÈLE D'IA

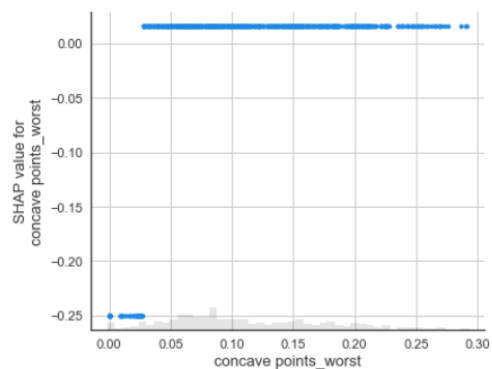
On constate sur ces visuels générés par **SHAP**, que les variables *concave points_worst* et *concavity_worst* sont primordiales, elles sont les entités les plus importantes qui expliquent le modèle et montrent leur contribution positive (rouge).

```
shap.plots.scatter(shap_values[:, "concavity_worst"])
```



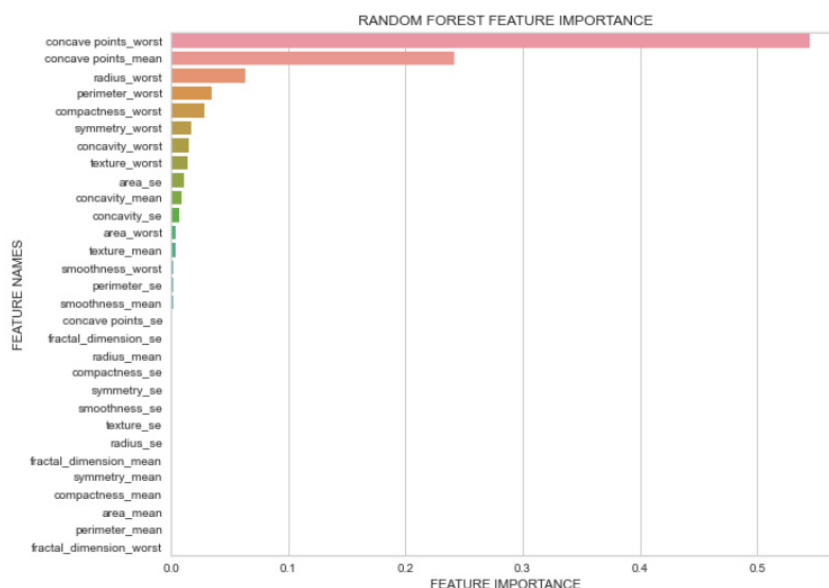
En analysant de plus près *concavity_worst* avec un *scatterplot*, on peut voir l'impact de cette variable : les valeurs faibles, entre 0 et 0,18, sont des prédicteurs plus négatifs que les valeurs entre 0,18 et 1, voir plus.

```
shap.plots.scatter(shap_values[:, "concave points_worst"])
```



Même constatation sur *concave points_worst* : si une valeur oscille entre 0 et 0,02, elle sera un prédicteur négatif, à contrario des valeurs entre 0,02 et 0,30.

Pour confirmer les variables les plus importantes je peux utiliser aussi la fonction *feature_importances_* de **Scikit-Learn**.



Le modèle est désormais prêt à être mis en production dans l'application.

DÉVELOPPEMENT DE L'APPLICATION

4.1 Front-end & back-end

L'architecture de l'application a été développée avec le framework, **Bootstrap**, une solution open-source pour la création de sites et d'applications web responsive dont les langages utilisés sont le *HTML*, le *CSS* ainsi que le *Javascript*.

Le modèle de *Machine Learning* a été intégré avec **Flask**, un micro-framework *Python*, très plébiscité grâce à ses nombreux avantages :

- Adapté au langage Python
- Facilement déployable en production
- Légèreté puisque "micro"
- Intégration simple des algorithmes de *Machine Learning* et *Deep Learning*

L'UX/UI design et le contenu de l'application répondent au cahier des charges :

- Respect des codes couleurs
- Clarté des informations
- Navigation simple
- Formulaire de prédiction avec condition de remplissage : valeur requise et obligatoire entre 0 et 1
- Interface d'identification des utilisateurs
- Interface d'enregistrement des nouveaux utilisateurs

Pink Breast
Votre nouvel outil en ligne pour prendre soin des autres.

En analysant les caractéristiques des noyaux cellulaires mammaires, obtenez un diagnostic sur la présence d'une tumeur maligne ou bénigne.

[Commencer](#)

Le cancer du sein

Le cancer du sein est le cancer le plus fréquent chez les femmes dans le monde. Il représente 25% de tous les cas de cancer et la tumeur plus de 32 millions de personnes en 2020. En France, l'Institut d'un programme national de diagnostic organisé afin d'être détecté précocement et d'en réduire le mortalité.

Le diagnostic

On peut s'interroger lors de la mammographie ou d'autres tests, mais il est difficile de savoir si une tumeur est bénigne ou maligne. Une biopsie est souvent effectuée pour confirmer le diagnostic. Le diagnostic est ensuite effectué au microscope et les caractéristiques des cellules sont analysées afin de déterminer si la tumeur est bénigne ou maligne.

L'analyse

L'analyse des cellules, par le biais des données, permet de prédire la présence d'une tumeur maligne ou bénigne. Notre modèle d'intelligence artificielle permet d'obtenir une aide aux professionnels de santé en permettant aux autres humains et en réduisant le temps d'analyse.

Application de prédiction

Remplissez les caractéristiques, en micromètres, du noyau cellulaire et obtenez une réponse rapide avec une précision de 99%.

Forme cancer - Au plus élevé	0
Forme cancer - Moyenne	0
Région - Au plus élevé	0
Région - Au plus bas	0
Compacité - Au plus élevé	0
Compacité - Au plus bas	0
Texture - Au plus élevé	0
Texture - Au plus bas	0
Région - Au plus bas	0
Compacité - Moyenne	0
Région - Au plus élevé	0
Texture - Moyenne	0

[Obtenir la prédiction](#)

Connexion à l'application

Identification [S'enregistrer](#)

[Valider](#)

Connexion à l'application

[Identification](#) **S'enregistrer**

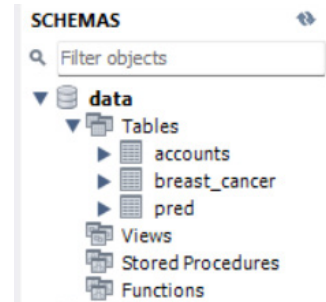
[Valider](#)

DÉVELOPPEMENT DE L'APPLICATION

4.2 La base de données MYSQL

Après développement de l'interface de l'application avec **Bootstrap** et **Flask**, il a fallu créer 2 nouvelles tables dans la base de données **MySQL** :

- pour sauvegarder les comptes utilisateurs
- pour sauvegarder les entrées du formulaire de prédiction



J'ai utilisé 2 librairies *Python* : **Flask_MySQLdb** et **SQLAlchemy** afin de faire communiquer la base de données et l'application.

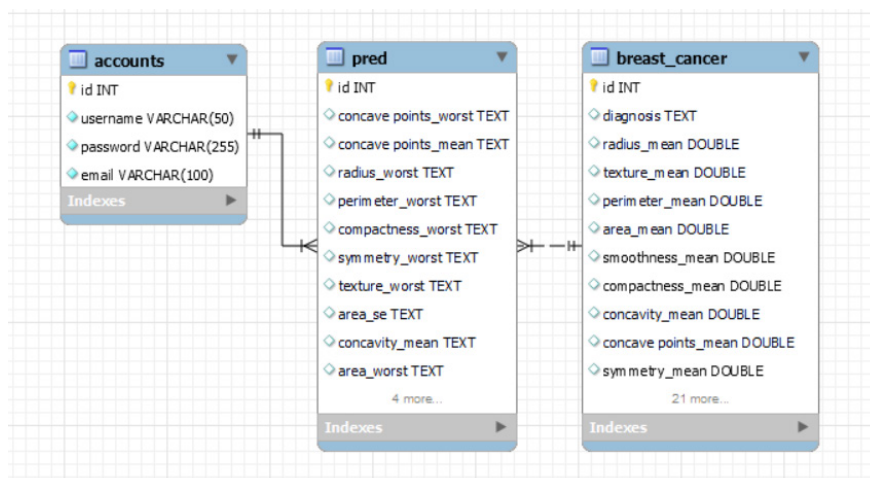
```
# create SQLAlchemy engine to connect to MySQL Database
engine = create_engine("mysql+pymysql://{user}:{pw}@{host}/{db}".format(host=hostname, db=dbname, user=username, pw=password))

# connect to the database
engine.connect()
```

Ainsi, dès qu'un utilisateur s'enregistre dans l'interface, ses données (identifiant, mot de passe, e-mail) sont sauvegardées dans la base de données *data*, table *accounts*.

Et dès qu'un utilisateur enregistré remplit et valide le formulaire de prédiction, les données sont sauvegardées dans la table *pred*.

Le schéma relationnel a été réalisé sur **MySQL Workbench**, le logiciel de gestion et d'administration de bases de données MySQL



DÉVELOPPEMENT DE L'APPLICATION

4.3 Tests unitaires, fonctionnels et de non régression

Les *tests unitaires* ont été réalisés avec la librairie **Pytest** :

- Vérification de l'affichage du `.predict()`
- Vérification de l'affichage correct du texte si `prediction = 0` et `prediction = 1`

```
(ENV) C:\Users\veron\Desktop\Soutenance\app\SMT GS Random Forest>pytest
===== test session starts =====
platform win32 -- Python 3.9.7, pytest-6.2.4, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\veron\Desktop\Soutenance\app\SMT GS Random Forest
plugins: anyio-3.5.0
collected 1 item

app_test.py .

===== 1 passed in 0.05s =====
```

Les *tests fonctionnels* n'ont détecté aucun dysfonctionnement de l'application.

Utilisateur	Objectif	Test fonctionnel
Nouvel utilisateur	S'enregistrer	OK
Utilisateur enregistré	S'identifier	OK
Utilisateur enregistré	Obtenir une prédiction	OK
Utilisateur enregistré	Cliquer sur un lien	OK
Administrateur	Récupérer les données dans la BDD	OK
Administrateur	Obtenir une prédiction	OK
Utilisateur enregistré	Ajout / modification d'un utilisateur	OK

Après amélioration du code et mise à jour, l'application fonctionne toujours correctement. Les *tests de non régression* assurent ainsi le bon comportement de l'application dans son ensemble malgré les modifications.

DÉVELOPPEMENT DE L'APPLICATION

4.4 Déploiement

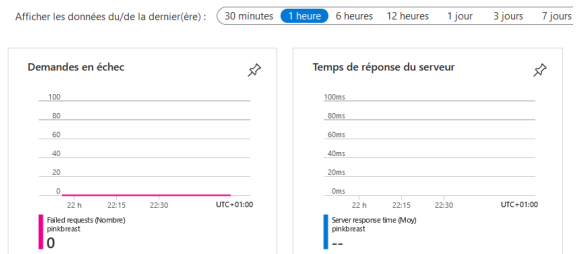
L'application a été déployée sur l'App Service d'**Azure** de **Microsoft**. Ce service permet de créer et déployer des applications web rapidement et facilement, en répondant à des performances de sécurité et conformité.

L'application est disponible en ligne à l'URL : <https://pink-breast.azurewebsites.net>

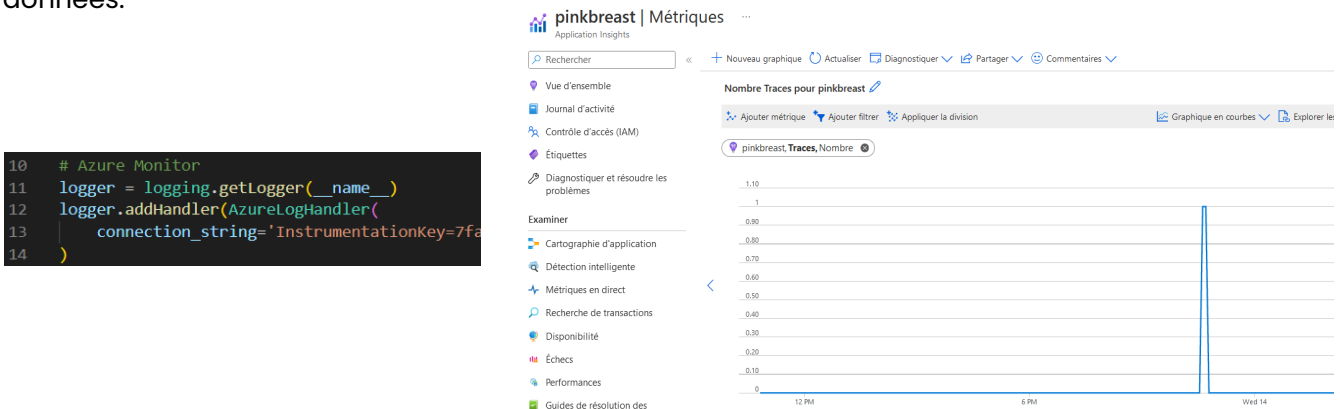
The screenshot shows the Azure portal interface for the 'pinkbreast' App Service. The top navigation bar includes 'Microsoft Azure', a search bar, and various icons. The left sidebar lists navigation options like 'Vue d'ensemble', 'Journal d'activité', 'Contrôle d'accès (IAM)', 'Étiquettes', 'Diagnostic et résoudre les problèmes', 'Microsoft Defender pour le cloud', and 'Événements (préversion)'. The main content area shows the 'Journaux' (Logs) tab, displaying a table of deployment events. The table has columns for 'Heure', 'ID de validation', 'Auteur de la validation', 'Statut', and 'Message'. A single deployment is listed for Wednesday, December 7, 2022, at 8:36:30 PM, with a status of 'Opération réussie (Actif)'.

4.5 Monitoring & alerte

Le monitoring de l'application est assuré grâce à l'**Application Insights**, une extension de **Azure Monitor** fournissant des fonctionnalités de surveillance des performances.



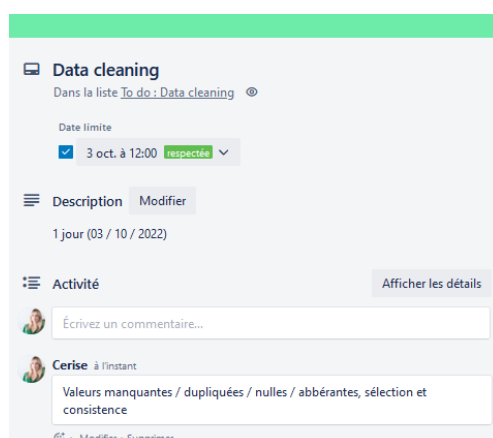
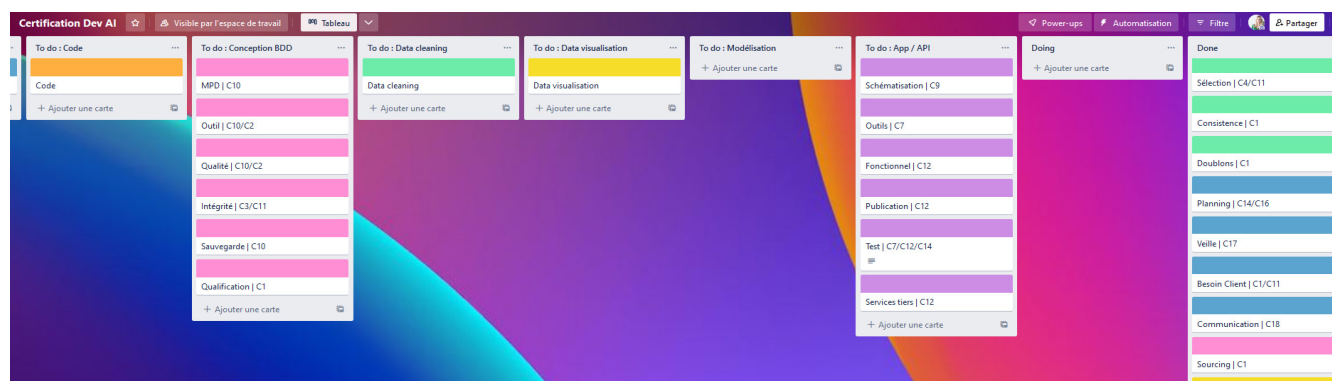
Le monitoring du modèle est aussi assuré par **Application Insights** grâce à la création d'une métrique personnalisée avec la librairie Python **OpenCensus**. La métrique correspond au *recall*. Cette librairie open-source recommandée par Microsoft, permet de collecter des données de télémétrie et de les exporter vers **Application Insights**. Une alerte intégrée, dès que le *recall* diminue sous le seuil de 0,85, à **Application Insights** été créé afin de surveiller que le comportement du modèle avec les nouvelles données.



GESTION DU PROJET

5.1 Planning

Le projet a été géré avec l'outil en ligne **Trello** : toutes les tâches ont été énumérées (*To Do / A faire*) puis triées en fonction des sprints effectués sur 3 mois (*Doing / En cours* - *Done / Terminé*).



Dans le détail de chaque tâches, une date limite a été définie.

- Préparation & veille
- Traitement data & IA
- Développement de l'application
- Tests & mise en production
- Rédaction

Octobre	Novembre	Décembre
● ● ● ●	● ● ● ●	● ● ● ●
	● ●	● ●

GESTION

DU PROJET

5.2 Communication & remerciements

Tout au long du projet, j'ai communiqué avec 2 experts : Mouchira Labidi, docteur en data science et Clothilde Celse, attachée de recherche clinique en cancérologie au Centre Léon Bérard à Lyon.

Egalement, j'ai eu le soutien de mes collègues de promotion de *Développeur en Intelligence Artificielle* afin de mener à terme ce projet, qui signe la fin d'un parcours de formation de 19 mois à leurs côtés. Plus d'une année très enrichissante où j'ai pu acquérir de nombreuses compétences, aussi en entreprise au sein de NEOS-SDI, plus particulièrement dans l'équipe de David Guillemin et Michaël Deschamps que je remercie particulièrement pour leur bienveillance et confiance.

CONCLUSION

Ce jeu de données ne contenait aucune données physiologiques des patientes, et c'est mon seul regret car il aurait été très intéressant de dresser des profils type par âge, conditions, mode de vie, métier, ect...

Il nen reste pas moins que c'est un sujet qui m'a tenu à coeur étant particulièrement sensible au domaine de la santé. La recherche et l'intelligence artificielle sont capables de véritables miracles et signe d'espoir.