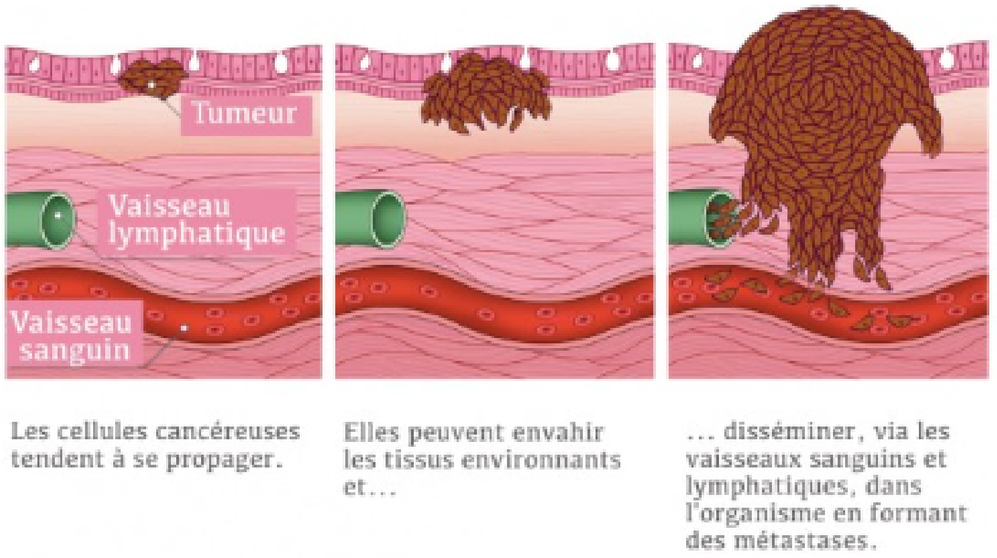
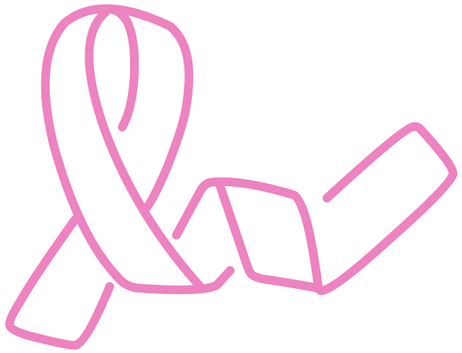
UTILISATION DU MACHINE LEARNING POUR LA DÉTECTION DU CANCER

KATHLEEN PERRIN & QUITTERIE LOISEAU

UTILISATION DU MACHINE LEARNING POUR LA DÉTECTION DU CANCER





CONTEXTE MÉDICAL

Le cancer est une maladie hétérogène, avec des milliers de sous-types génétiques et phénotypiques. Les traitements standards ne suffisent pas toujours, d'où le besoin de médecine personnalisée

UTILISATION DU MACHINE LEARNING POUR LA DÉTECTION DU CANCER

RÔLE DU MACHINE LEARNING

- **Analyse de données massives** issues du séquençage génomique, transcriptomique et protéomique.
- **Détection de motifs cachés** dans les données biologiques pour identifier des biomarqueurs.
- **Prédiction de la réponse** aux traitements en fonction du profil moléculaire du patient.
- **Aide au diagnostic** par classification automatique d'images médicales (histopathologie, radiologie).

UTILISATION DU MACHINE LEARNING POUR LA DÉTECTION DU CANCER

APPLICATIONS CONCRÈTES

- **Stratification des patients** : regrouper les malades selon des signatures génétiques pour adapter les thérapies.
- **Découverte de nouvelles cibles** thérapeutiques grâce à l'analyse statistique des réseaux de gènes.
- **Optimisation des essais cliniques** en sélectionnant les patients les plus susceptibles de répondre.
- **Imagerie médicale :** utilisation de réseaux de neurones pour détecter des anomalies invisibles à l'œil humain.

UTILISATION DU MACHINE LEARNING POUR LA DÉTECTION DU CANCER

DÉFIS TECHNIQUES

- **Qualité et quantité** des données (bruit, biais, données manquantes).
- **Interprétabilité des modèles** (boîtes noires vs modèles explicables).
- **Intégration multidimensionnelle** (génome, épigénome, environnement).
- **Collaboration** entre médecins, biologistes et informaticiens.

PERSPECTIVES

- Vers une **médecine de précision** où chaque patient reçoit un traitement adapté à son profil.
- **Développement de modèles prédictifs** robustes pour anticiper l'évolution de la maladie.
- **Utilisation de l'IA** pour accélérer la recherche de médicaments et réduire les coûts.

UTILISATION DU MACHINE LEARNING POUR LA DÉTECTION DU CANCER

PROBLEMATIQUE

Comment concilier la performance prédictive du Deep Learning avec l'exigence d'explicabilité médicale pour fiabiliser le diagnostic du cancer ?

OBJECTIF

Développer un modèle d'IA capable de classifier les tumeurs (Bénignes vs Malignes) à partir de caractéristiques cellulaires.

DÉFI TECHNIQUE

Dépasser la "boîte noire" (Black Box) en expliquant pourquoi l'IA prend une décision (via Integrated Gradients).

MÉTHODOLOGIE ET ARCHITECTURE TECHNIQUE

SOURCE GITHUB

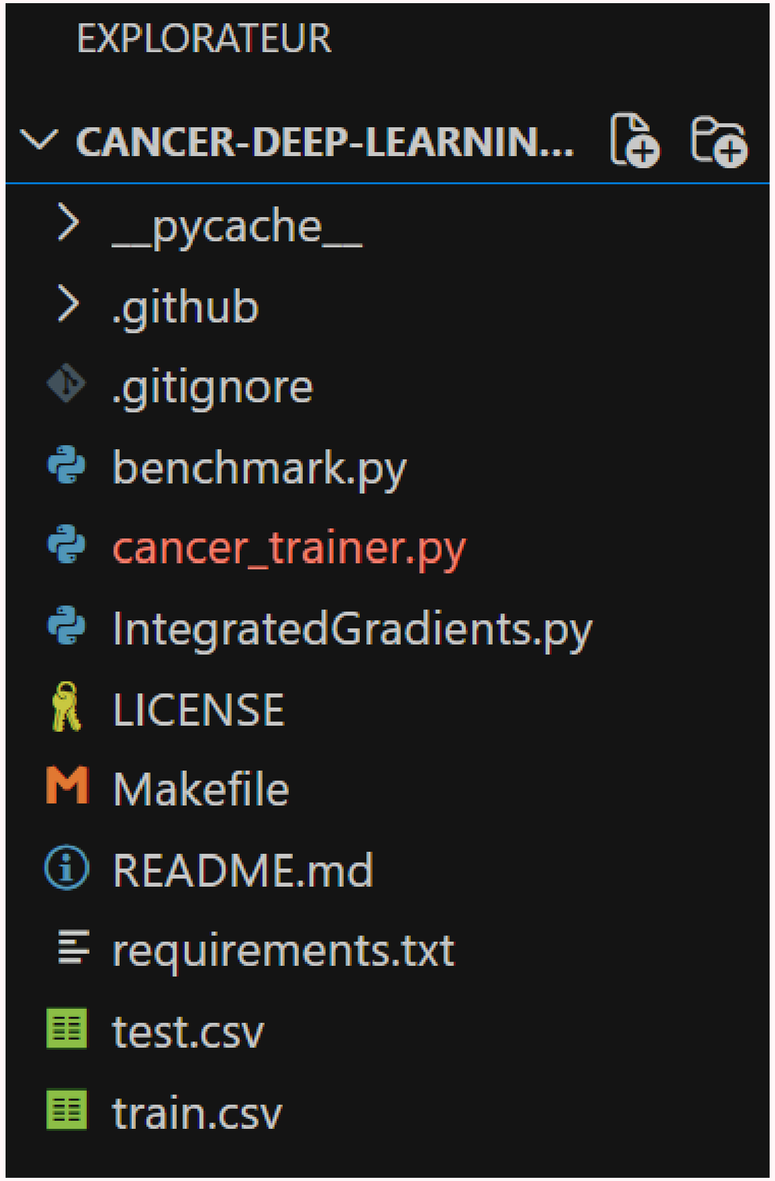
Réseau neuronal profond de Keras utilisant des données sur le cancer du sein avec explication des prédictions
Auteur: Mark-Watson

1. **cancer_trainer.py** : Construction et entraînement du réseau de neurones (Deep Learning).

2. **IntegratedGradients.py** : Module d'explicabilité pour interpréter les décisions

3. **benchmark.py** : Script version 'Haute Capacité' du réseau de neurones

4. **test.csv & train.csv** : Jeux de données cliniques partitionnés pour l'apprentissage et l'évaluation



LES DONNEES D'ENTREE

1	5,4,4,9,2,10,5,6,1,1
2	10,10,10,4,8,1,8,10,1,1
3	8,2,4,1,5,1,5,4,4,1

Chaque ligne représente **un patient** avec **9 valeurs** numériques mesurées en laboratoire

Position	Caractéristique	Valeur	Echelle
1	épaisseur des amas cellulaires	5	1 à 10
2	uniformité taille	4	1 à 10
3	uniformité forme	4	1 à 10
4	adhésion cellulaire	9	1 à 10
5	taille cellules	2	1 à 10
6	noyaux nus visibles	10	1 à 10
7	chromatine	5	1 à 10
8	Normal nucléoles	6	1 à 10
9	Mitoses (divisions cellulaires)	1	1 à 10
10	Classe cible	1	0=bénin, 1=malin

COMMENT CES DONNÉES SONT OBTENUES ?

1. Un médecin fait une biopsie (prélèvement de tissu)
2. Il observe les cellules au microscope
3. Il note chaque caractéristique sur une échelle de 1 à 10 :
    - 1 = très normal
    - 10 = très anormal

Ces 9 chiffres deviennent les données d'entrée du modèle

ARCHITECTURE DU RÉSEAU DE NEURONES

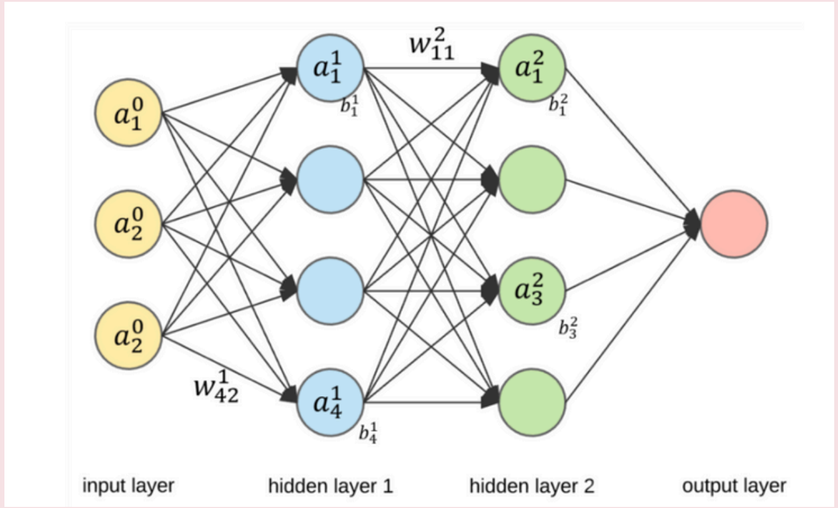
Voici l'architecture de notre réseau de neurones, construite avec la librairie Keras

```
X_train = np.array([[float(j) for j in i.rstrip().split(",")]
                    for i in open("train.csv").readlines()])
Y_train = X_train[:,-1]
X_train = X_train[:,0:-1]

X_test = np.array([[float(j) for j in i.rstrip().split(",")]
                    for i in open("test.csv").readlines()])
Y_test = X_test[:,-1]
X_test = X_test[:,0:-1]

inputs = Input(shape=[9])

x = Dense(64, activation='relu')(inputs)
x = Dense(64, activation='relu')(x)
probs = Dense(1, activation='sigmoid')(x)
```



→ entrée : Les 9 biomarqueurs

→ traitement : 2 couches de 64 neurones avec une activation ReLU. C'est ici que l'IA détecte les corrélations complexes entre les symptômes.

↓ sortie : La dernière couche transforme le tout en une probabilité unique : est-ce malin ou bénin ?

- 497 patients : train
- 151 patients : test

APPRENTISSAGE & EXPLICABILITÉ

```
model1 = Model(inputs=inputs, outputs=probs)
model1.compile(optimizer='sgd', loss='binary_crossentropy')
💡
model1.fit(X_train, Y_train, epochs=1000, batch_size=128,
           validation_split=0.15, verbose=True)

int_gradients = integrated_gradients(model1)
```

binary_crossentropy : fonction mathématique qui calcule l'erreur entre la prédiction de l'IA et le diagnostic réel du médecin.

L'Entraînement (fit) : Le modèle s'entraîne sur 1000 cycles (**epochs**).

- 15% des données pour se tester
- d'entraîner sur les 85% restants.

IntegratedGradients.py

Ouvrir la "Boîte Noire" par les Mathématiques

Objectif : Attribuer un score d'importance à chaque biomarqueur dans la décision finale.

- calcule la sensibilité du modèle (la dérivée) à chaque micro-étape de l'évolution de la maladie.
- intégration mathématique de toutes ces sensibilités

```
# Compute gradients for all interpolated samples
explanation = []
for i in range(len(samples)):
  gradients = self._compute_gradients(samples[i], outc)
  _temp = np.sum(gradients, axis=0)
  explanation.append(np.multiply(_temp, step_sizes[i]))
```

RÉSULTATS

- Échantillon BÉNIN (pas de cancer)

```
** Contributions to classification for sample type  benign sample  **
        Clump Thickness :       -72
        Uniformity of Cell Size :       14
        Uniformity of Cell Shape :      8
        Marginal Adhesion :     -12
        Single Epithelial Cell Size :   -100
        Bare Nuclei :   -18
        Bland Chromatin :       -66
        Normal Nucleoli :       -5
        Mitoses :       -27
```

Les 3 facteurs les plus importants qui ont dit "c'est bénin" :

- **Single Epithelial Cell Size** : -100 (facteur décisif)
- **Clump Thickness** : -72 (très important)
- **Bland Chromatin** : -66 (important)

Ces cellules ont des caractéristiques normales, donc pas de cancer

- Échantillon MALIN (cancer)

```
** Contributions to classification for sample type  malignant sample  **
        Clump Thickness :       74
        Uniformity of Cell Size :       -13
        Uniformity of Cell Shape :      22
        Marginal Adhesion :     12
        Single Epithelial Cell Size :   -22
        Bare Nuclei :   100
        Bland Chromatin :       75
        Normal Nucleoli :       78
        Mitoses :       -1
```

Les 3 facteurs les plus importants qui ont dit "c'est un cancer" :

- **Bare Nuclei :** 100 (noyau cellulaire anormal)
- **Normal Nucleoli**: 78 (Nucléoles)
- **Bland Chromatin** : 75 (ADN se duplique rapidement)

Ces anomalies sont des signes de cancer

Benchmark.py modifé

- Le fichier original proposait un réseau **"Mammouth" (2 couches de 1000 neurones)**.

- **Problème** :

Sur un petit dataset médical, modèle est "sur-dimensionné" (**Overkill**) et risque d'apprendre par cœur sans généraliser.

=> Passage de Keras (Deep Learning) à Scikit-Learn (Machine Learning Classique).

Intégration de **Random Forest (Arbres décisionnels)** et **SVM (Vecteurs de support)**

Objectif :

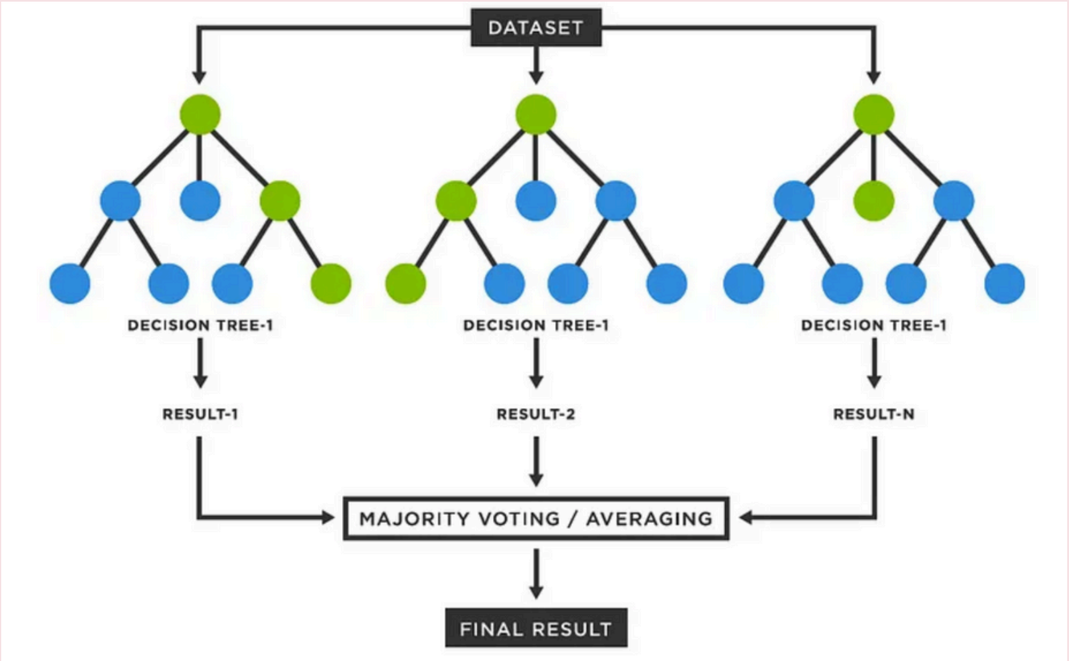
- Valider si la complexité du Deep Learning est vraiment nécessaire.
- Comparer la métrique critique : le Rappel (Recall)

A-t-on vraiment besoin d'une IA complexe de type Deep Learning pour ce diagnostic, ou une méthode classique plus robuste suffit-elle ?

COMPARAISON

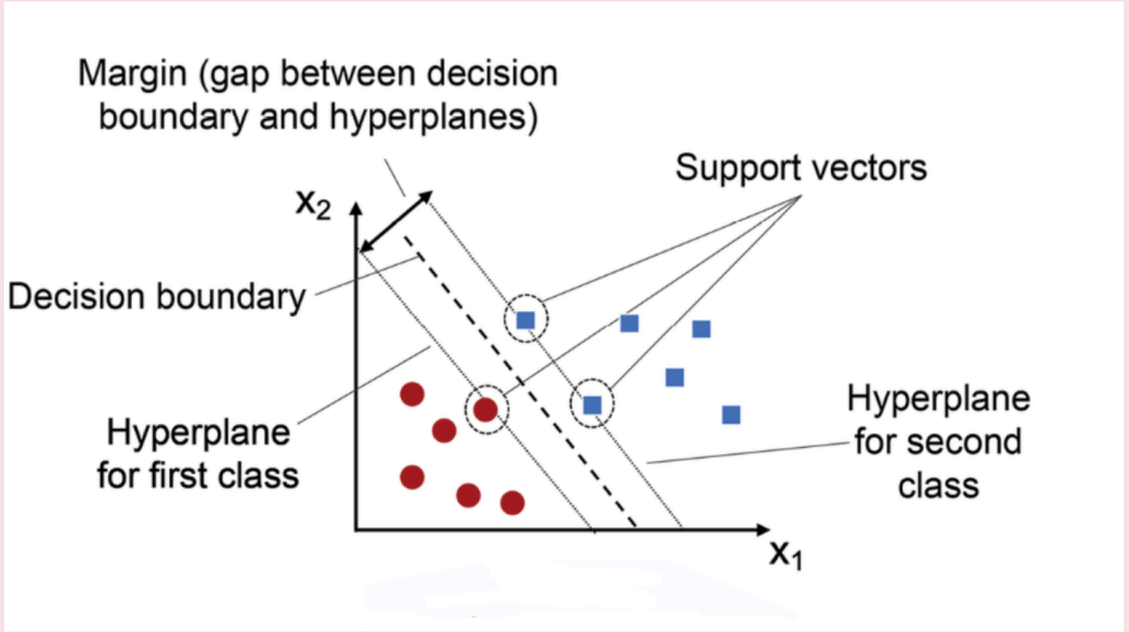
Random forest

Au lieu de chercher un seul modèle parfait, l'algorithme construit une multitude d'Arbres de Décision faibles et décorrélés.



SVM (Support Vector Machine)

Le SVM cherche la séparation la plus "large" possible.



Résultats

```
⏳ Entraînement de : Random Forest...
📊 RÉSULTATS : Random Forest
   > Précision (Accuracy) : 95.36%
   > Rappel (Sécurité)    : 94.00%
   > Faux Négatifs        : 3 (Malades non détectés)
------------------------------

⏳ Entraînement de : SVM (Classique)...
📊 RÉSULTATS : SVM (Classique)
   > Précision (Accuracy) : 96.69%
   > Rappel (Sécurité)    : 100.00%
   > Faux Négatifs        : 0 (Malades non détectés)
------------------------------
```

Random forest

- bonne précision globale (95%),
- Faux Négatifs : 3 patients malades.

En termes cliniques, c'est inacceptable

Pourquoi ?

Découpage de manière orthogonale.

Imprécisions sur les données limites

SVM

- détecté o faux négatif.
- modèle le plus sécuritaire

Pourquoi ?

Mathématiquement conçu pour la généralisation.

Il réduit le risque de sur-apprentissage (overfitting).

Conclusion

Modèle	Type	Précision (Accuracy)	Sécurité (Recall)	Verdict
Deep Learning	Réseau de Neurones	97%	96%	**Le plus Explicable** (Grâce aux gradients)
Random Forest	Arbres de Décision	95%	94%	**Trop risqué** (3 malades ratés)
SVM	Géométrique	**97%**	**100%**	**Le plus Sûr** (0 malade raté)

- Le Deep Learning offre la modélisation la plus fine et explicable.
- Le SVM offre la sécurité maximale.
- Le Random Forest est ici le moins adapté car trop risqué."

Dans un contexte hôpital, nous privilégierons le SVM pour le tri initial (pour ne rater personne), couplé à notre module Deep Learning pour l'explication du diagnostic au médecin.

Sources

- **Github** : https://github.com/mark-watson/cancer-deep-learning-model/blob/master/cancer_trainer.py
- **SVM:** https://vitalflux.com/classification-model-svm-classifier-python-example/
- **Random Forest:** https://datasciencedojo.com/blog/random-forest-algorithm/
- **Réseaux de neurones multicouches :** https://www.aspexit.com/reseau-de-neurones-on-va-essayer-de-demystifier-un-peu-tout-ca-1/