

# Doctor:Hadaca

- Triki Bilelle
- Cornede Cédric
- Cintra Paul
- Abichou Asma
- Amrane Lydia
- Nait Larbi Takfarinas



15.000.000 décès par an

Les cancers qui tuent le plus sont les cancers du poumon, du foie et de l'estomac.

Depuis 1989, en France, les cancers sont la première cause de décès chez l'homme.

Entre 1980 et 2005, le nombre de cancers a augmenté de 90 %



**CANCER**



**SOLUTION**

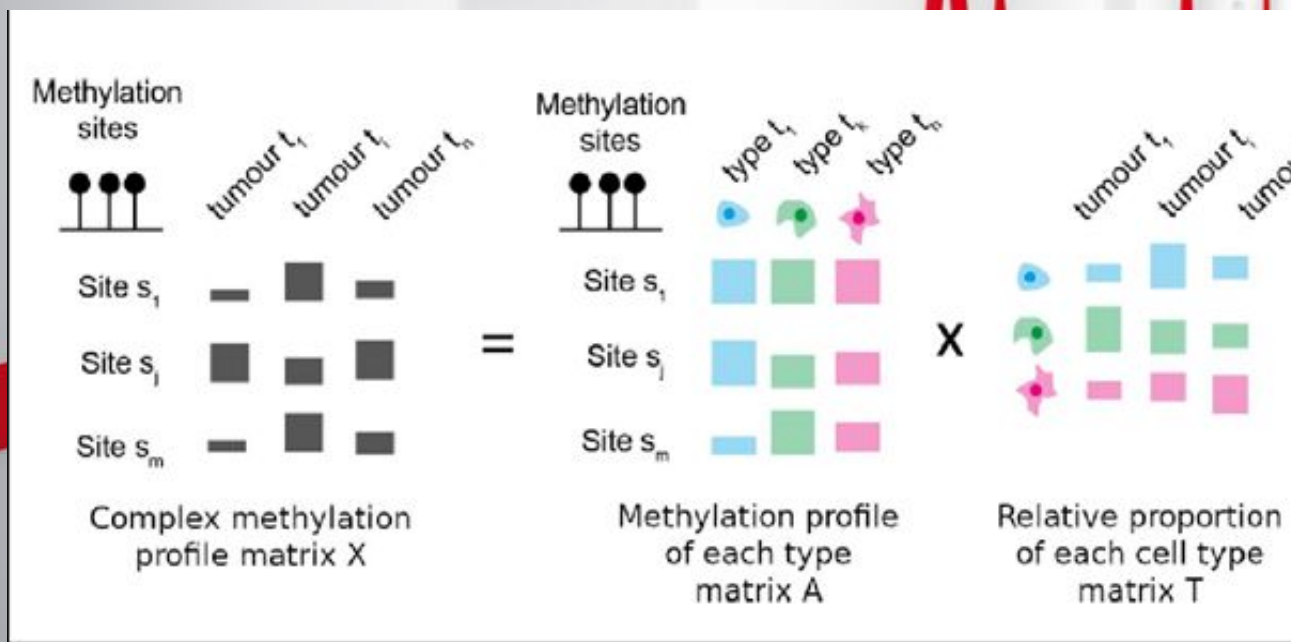
**HADACA  
Health Data  
Challenge**







# Étape 1: Preprocessing Prétraitement des données pour les transformer en un format qui sera traité plus facilement et efficacement par le modèle d'apprentissage

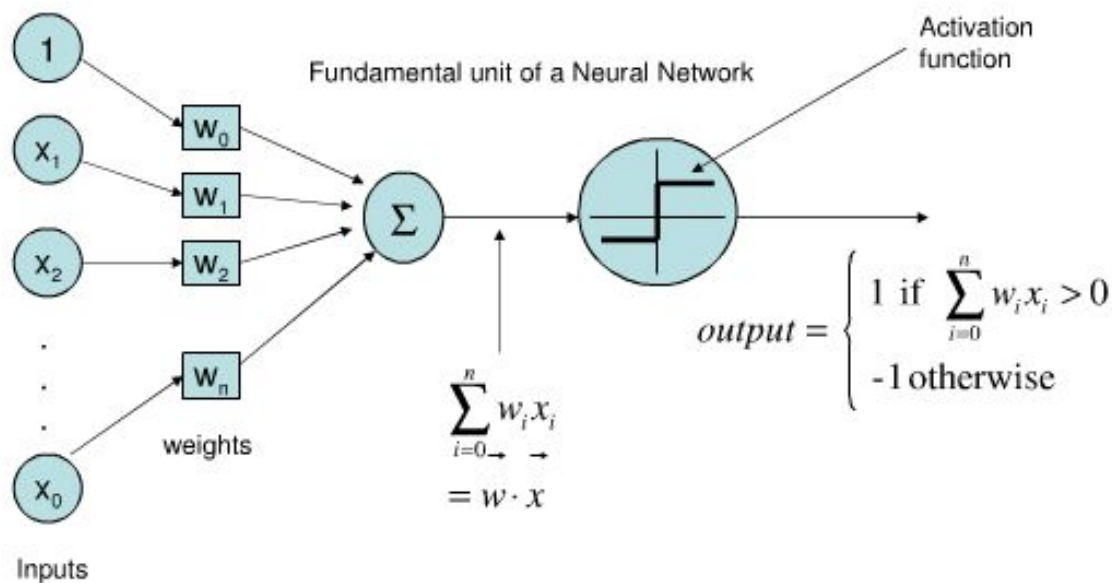


## Pseudo-code pour le preprocessing :

```
Factorisation-mat (matriceBrute[x][y], t) {  
  nb_feature = TruncatedSVD(n_components =  
  t)  
  data-reduce =  
  svdfit_transformation(matriceBrute[x][y])  
  variance = svdcomponent(matriceBrute[x][y])  
  D-matrix = np.dot(data-reduce, variance)  
  calculer le taux d'erreurs entre matriceBrute  
  et D-matrix  
  retourner data-reduce, variance  
}
```

```
// matriceBrute[x][y] c'est la matrice qui  
contient les données brutes & t est le nombre  
de patients (  
// data-reduce c'est la matrice de factorisée  
// np.dot c'est pour faire le produit entre x et y  
// variance est la matrice qui s'occupe de  
stocker la variance max de chaque donnée
```

## Étape 2: Preprocessing Améliorer le taux de classification en classifiant les données qui viennent d'être traitées



labels[]	# toutes les classes différentes
images[] {..	{ #toutes les images
	#on part du principe que plus la distance entre une image

...}	#et un label est grande plus la probabilité que cette image appartienne
	#à ce label est grande
res = 0 ;	#nombre d'image bien classées
for i in images[]:	# pour parcourir toutes les images
bestdist=0	# distance max entre image et un label
predictedLabel=0	# le label ayant la plus grande distance avec l'image
for j in labels[]:	# parcourir toute les labels ( les classes )
if dist(images[i],labels[j])>bestdist:	# On cherche à trouver la distance max par image
bestdist=dist(images[i],[labels[j]	
predictedLabel=labels[j]	
if(predictedLabel==images[i].trueLabel):	# si label prédit , est égale au vrai label de l'image on
res=res+1	#augmente le résultat de
print("efficacité de " (res/images[].length)*100 "%")	1 .
	#le pourcentage de réussite !



## Étape 3: Affichage Représentation graphique des données

Figure 5 : Histogramme nous montrant la quantité de chaque composant en fonction du nombre de patients

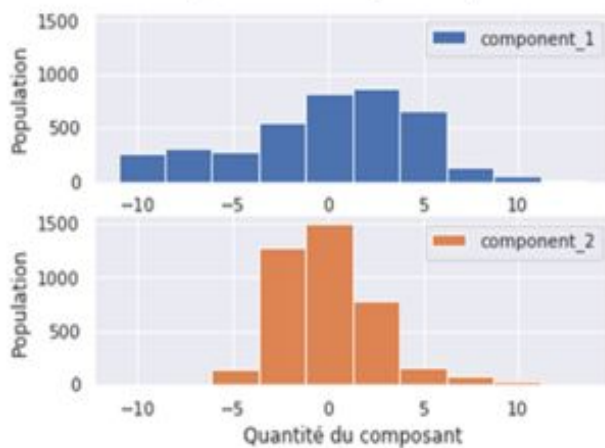
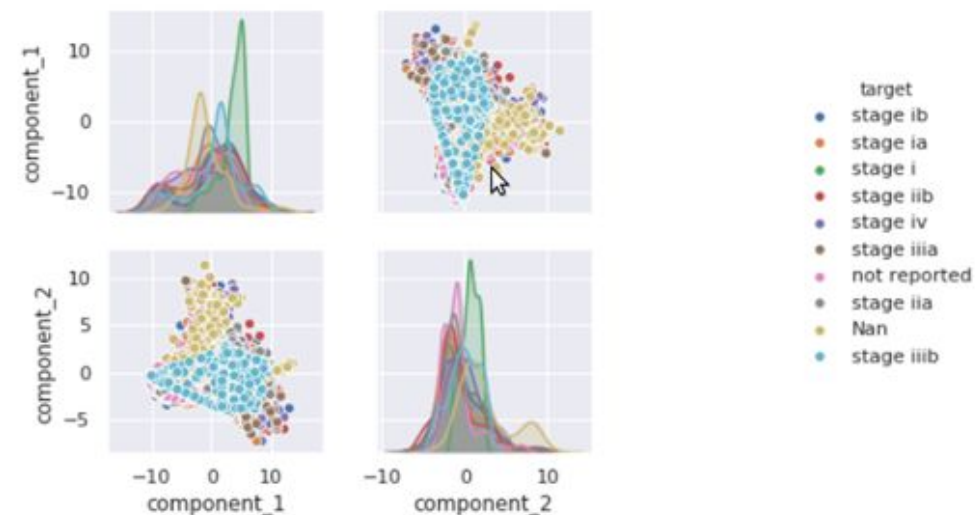


Figure 6: Pairplots montrant la corrélation entre les composants et le stade de cancer atteint ainsi que le lien entre les deux composants





Success.