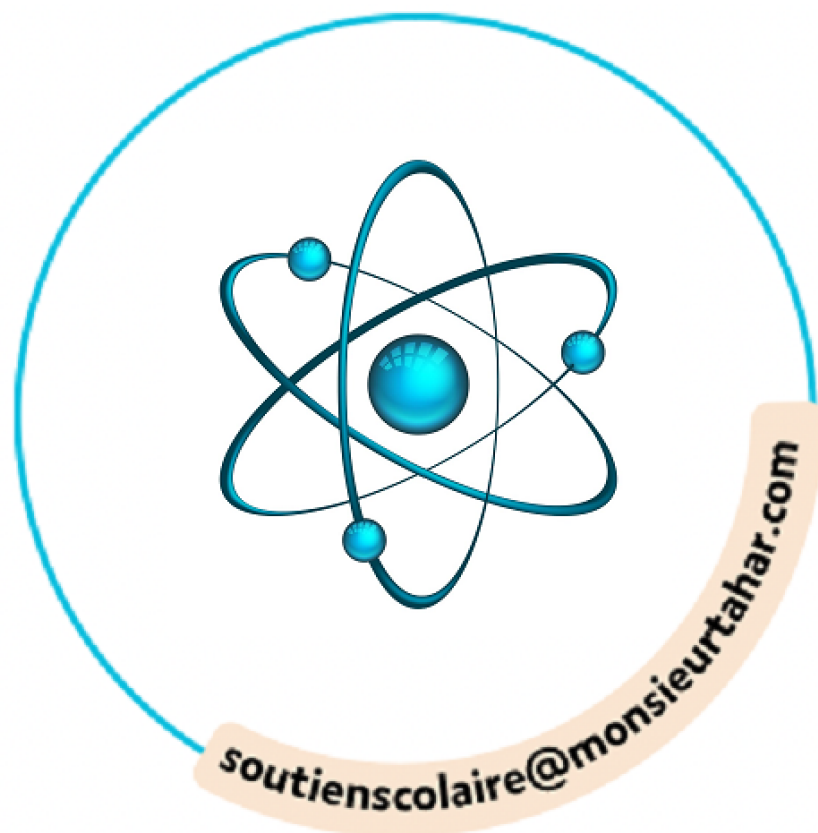
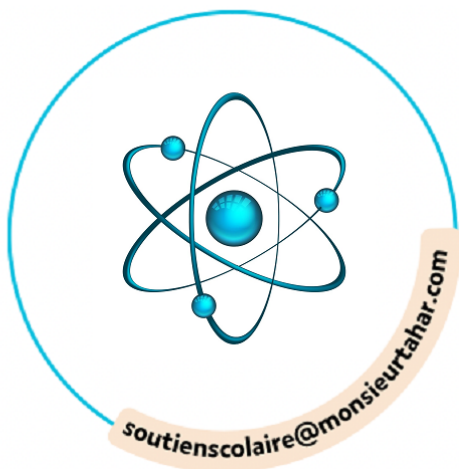


ENSEIGNEMENT SCIENTIFIQUE



PHYSIQUE -CHIMIE

CHAPITRE 7



Exercices

Tester ses connaissances

QCM

1. c. / 2. b. ET c. / 3. a. ET d.

4 Vrai ou Faux ?

a. Faux. / b. Faux. / c. Vrai. / d. Vrai.

5 Phrases à construire

- Une **image** numérique est composée de **pixels**, chaque pixel affiche une **couleur** codée sur trois **octets**.
- Un **exécutable** est un **type** de fichier contenant des **informations** permettant de générer des **instructions**.
- Pour certaines tâches, utiliser la **programmation** pour **coder** un **algorithme** est inefficace, il convient alors d'**entraîner** un **modèle d'IA**.

6 Définitions inversées

- Faux-positif
- Fréquence d'échantillonnage
- Machine Learning*. *Deep learning* est valable aussi, même s'il s'agit d'une sous-catégorie du *machine learning*.
- Bug

7 Faux positifs

1. Tableau de contingence rempli :

	Étoile	Rond
Bien classifié	15	27
Mal classifié	4	3

2. On veut sélectionner les étoiles, les faux positifs sont donc les ronds, identifiés à tort comme des étoiles : il y en a 3.

8 Calculs !

1. Rappel : Dans l'activité 1, on a vu que 400 pages de textes correspondaient à 0,9 Mo.

$15 \times 10^6 \times 0,9 \times 10^6 = 13,5 \times 10^{12} = 13\,500 \times 10^9$ octets, il faudra donc 13 500 Go d'espace de stockage.

2. On calcule l'espace occupé par une seconde d'une vidéo 720 p au format 16:9 :

$$720 \times \frac{16}{9} = 1\,280 ; \text{ ainsi } 720 \times 1\,280 \times 3 \times 24 = 66\,355\,200 \text{ octets.}$$

On divise ensuite l'ensemble de ce qui est stocké à la BNF :

$$\frac{13\,500 \times 10^9}{66\,355\,200} \approx 230\,450,5 \text{ secondes. On convertit en heures/minutes/seconde : 64 heures et 50 secondes.}$$

3. $44\,100 \times 2 = 88\,200$ octets pour une seconde.

$88\,200 \times 60 \times 60 = 31\,752\,000$ octets = 317,5 Mo pour une heure.

$$\frac{13\,500 \times 10^9}{317,5 \times 10^6} = 52\,519,7 \text{ heures, c'est-à-dire 52\,519 heures et 42 minutes.}$$

Exercices

Développer ses compétences

10 Différents symptômes pour une même erreur

1. L'algorithme affiche 17 avec la première liste ; 20 avec la deuxième liste ; 14 pour la troisième liste et un message d'erreur pour la dernière liste.

Si l'on calcule « à la main » ces sommes : on trouve 17 ; 20 ; 18 et 20.

2. Les résultats diffèrent à cause des longueurs des listes. L'algorithme ne prend en compte que les cinq premiers termes de la liste et les additionne. Néanmoins, pour L2, cela donne le bon résultat car le terme manquant est un 0, ce qui n'est pas le cas pour L3.

3. Algorithme modifié :

```
def total_animaux(L):
    n=len(L)
    S=0
    for k in range(n):
        S=S+L[k]
    return S
```

11 Arbre de décision

Fonction en Python :

```
def badminton(meteo,temperature,vent):
    if meteo=="pluie":
        return "NON"
    elif temperature>30:
        return "NON"
    elif vent=="fort":
        return "NON"
    else:
        return "OUI"
```

12 L'apprentissage

1. Les éléments communs sont :

- l'essai-erreur ;
- le retour d'information sur la justesse de la proposition ;
- l'ajustement progressif.

2. Les données sont la position et la couleur de chaque pion. L'apprentissage est la reproduction du code. L'ajustement à la réponse est le retour d'information, qui permet à « l'apprenant » de corriger sa réponse (son « modèle »).

13 Le bug du Zune

1. Dans les deux premières lignes, on définit les variables Année et Jour.

On a ensuite une boucle conditionnelle « Tant que Jour > 365 ». L'algorithme entrera dans cette boucle par exemple lorsque la variable Jour vaudra 366 (ce qui est le cas du dernier jour d'une année bissextile, mais aussi du 1er janvier suivant une année non bissextile) ou 367 (ce qui est le cas du 1er janvier suivant une année bissextile).

Dans cette boucle, on distingue deux conditions selon que l'année qui se termine est bissextile ou non et dans chacun des cas, on ramène la variable Jour à 1 et on augmente Année de 1 (pour indiquer qu'on est le 1er janvier de l'année suivante).

2. Avec le 242^{ème} jour d'une année non bissextile ou le 345^{ème} jour d'une année bissextile, l'algorithme n'entrera jamais dans la boucle « Tant que (Jour > 365) », il ne se passera donc rien. Les variables Année et Jour resteront identiques à l'issue de cet extrait de code et ce seront les instructions qui suivent cet extrait (après la boucle) qui seront effectuées.

3. Le 366^{ème} jour d'une année bissextile, l'algorithme entre dans la boucle « Tant que (Jour > 365) ». À l'intérieur de cette boucle, c'est la première condition qui est vérifiée (« Si Année est bissextile »). Mais, dans cette condition, le seul cas envisagé est « Si jour > 366 » (c'est-à-dire le jour 367, autrement dit le 1er janvier de l'année suivante). Or, à ce moment le jour vaut 366. Il ne se passe donc rien et le jour reste égal à 366, la boucle recommence alors puisque la condition « Jour > 365 » est toujours vérifiée et ainsi de suite. La boucle est alors exécutée indéfiniment, d'où le problème.

Une modification pour éviter cette erreur serait de programmer une sortie forcée de la boucle si jamais on entre dans le test de l'année bissextile, mais pas dans celui du Jour > 366 :

```
Année= année en cours
Jour= numéro du jour en cours
Tant que (Jour > 365) :
    Si Année est bissextile :
        Si Jour > 366 :
            On enlève 366 à Jour
            On augmente Année de 1
        Sinon :
            Arrêter la boucle
    Sinon :
        On enlève 365 à Jour
        On augmente Année de 1
```

14 Test de primalité

1. On peut par exemple tester ces programmes sur tous les entiers compris entre 2 et 9. Attention, 2 est un cas très spécial car c'est le seul nombre pair qui est tout de même premier. Quant à 9, c'est le plus petit nombre impair qui n'est pas premier.

2. L'idée de ces trois programmes est à peu près la même. On teste les nombres plus petits que n . Dès que l'un d'entre eux divise n , on peut dire que n n'est pas premier. Cependant, si aucun d'entre eux ne divise n , c'est que n est premier.

a) Le programme A donne un résultat faux avec $n = 9$. En effet, il renvoie True avec tous les nombres impairs. Le problème vient du fait que, dans la boucle, on teste d'abord la divisibilité de n par 2 ; si n est pair, alors le programme renverra False (ce qui est correct), mais SINON (c'est-à-dire pour n'importe quel nombre impair), le programme renverra alors True (ce qui est correct pour 3, 5 et 7, mais pas pour 9).

Remarque : pour $n = 2$, ce programme ne renverra rien du tout car $\text{range}(2,n)$ est une liste vide vu que n est exclu.

b) Le programme B donne un résultat faux avec $n = 2$. En effet, il renverra toujours False, quel que soit le nombre n entré supérieur ou égal à 2. Le problème vient du fait qu'on commence la boucle avec $k = 1$. Or tous les entiers sont divisibles par 1 donc la condition « $n\%k==0$ » sera vérifiée dès le début.

c) Le programme C donne également un résultat faux avec $n = 2$. En effet, il renverra toujours False, quel que soit le nombre n entré. Le problème vient du fait que, dans la boucle, k prend toutes les valeurs jusqu'à n INCLUS. Or tout entier est divisible par lui-même donc la condition « $n\%k==0$ » sera vérifiée lorsque k vaudra n .

Une version corrigée de ce programme est :

```
def primalite_correct(n):
    for k in range(2,n):
        if n%k==0:
            return False
    return True
```

3. Le modèle ne pourra interpréter que les données en entrée de son modèle. Le panneau stop n'en faisant pas partie, il sera ignoré, la situation sera traitée comme un virage normal.

4. Soleil → pique-nique : [1,0]→[1,0]
Pluie → cinéma : [0,1]→[0,1]

Poids	Valeur
P_11	1
P_12	0
P_21	0
P_22	1

16 Précision ou automatisation ?

La précision et l'automatisation sont deux paramètres en vases communicants (hors programme, mais derrière cela se cache la notion de précision et de rappel https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel).

Le critère discriminant est souvent la criticité de la tâche, qui détermine à quel point une erreur est grave ou non.

Détection de spams : automatisation.

Diagnostic d'un cancer : précision.

Identification faciale pour déverrouiller un téléphone portable : précision.

Compréhension d'une commande vocale pour une enceinte connectée : automatisation.

17 Résultats d'une recherche

1. Avec Jacques :

	Jacques	Pas Jacques
Jean-Jacques Goldman	3	1
Pas Jean-Jacques Goldman	2	2

Avec Gold :

	Gold	Pas Gold
Jean-Jacques Goldman	4	0
Pas Jean-Jacques Goldman	2	2

Avec Goldman :

	Goldman	Pas Goldman
Jean-Jacques Goldman	4	0
Pas Jean-Jacques Goldman	0	4

2. Certaines chaînes de caractère possibles sont « man » ou « man - » par rapport à la série de musiques données pour cette recherche.

Avec man :

	man	Pas man
Jean-Jacques Goldman	4	0
Pas Jean-Jacques Goldman	0	4

18 Efficacité d'un test médical

1. Tableau de contingence associé au test :

	Test positif	Test négatif
Personne malade	0,042	0,008
Personne saine	0,14	0,81

2. Le nombre de personnes avec un résultat positif est : $500 \times 0,8 = 400$ personnes. Or la probabilité, pour une personne dont le test est positif, d'être vraiment malade est le nombre de malades avec un test positif divisé par le nombre total de personnes dont le test est positif, c'est-à-dire :

$$\frac{0,042}{0,042 + 0,14} \approx 0,23.$$

On peut donc estimer que, parmi les 400 personnes dont le test est positif, il y a $400 \times 0,23 = 92$ personnes malades.

Cependant, il peut aussi y avoir des personnes malades parmi celles dont le test est négatif. Il y a 100 personnes dont le test est négatif. Or la probabilité, pour une personne dont le test est négatif, d'être quand même malade est le nombre de malades avec un test négatif divisé par le nombre total de personnes

dont le test est négatif, c'est-à-dire $\frac{0,008}{0,008 + 0,81} \approx 0,01$. On

peut donc estimer que, parmi les 100 personnes dont le test est négatif, il y a $100 \times 0,01 = 1$ personne malade.

En tout, on peut donc estimer qu'il y a $92 + 1 = 93$ malades parmi les 500 personnes testées.

19 L'IA, un modèle à évaluer

1. Tableau de contingence :

	Détecté comme chien	Détecté comme autre
Chien	7	3
Autre	2	8

2. Pour obtenir la fiabilité, il faut diviser le nombre de photos avec un chien bien reconnues par le nombre total de photos pour lesquelles l'IA a affirmé qu'il y avait un chien.

La fiabilité de cette IA est donc $\frac{7}{2+7} \approx 78$, c'est-à-dire 78 %.

3. Nouveau tableau de contingence :

	Détecté comme chien	Détecté comme autre
Chien	7	3
Autre	3	8

4. On effectue le même calcul qu'à la question 2 mais avec les nouvelles données.

La nouvelle fiabilité est donc $\frac{7}{3+7} = 0,7$, c'est-à-dire 70 %.

20 EXERCER SON ESPRIT CRITIQUE

Une IA judiciaire est-elle forcément impartiale ?

Estimation de l'efficacité de l'algorithme :

- taux de réussite de prédiction d'une récidive assez proches pour les deux groupes ethniques : 0,63 et 0,59 ;
- taux de réussite de prédiction d'une non-récidive assez proches pour les deux groupes ethniques : 0,65 et 0,71.

L'efficacité n'est pas très bonne (entre 30 et 40 % d'échec), mais elle est à peu près équivalente dans les deux groupes ethniques bien qu'elle montre une différence de 4 à 6 % en faveur du groupe B. Cependant, les critères de fiabilité de l'algorithme sont problématiques :

- sensibilité (capacité à prédire correctement la récidive) meilleure pour le groupe A (0,72 vs 0,52) ;
- spécificité (capacité à prédire correctement la non-récidive) meilleure pour le groupe B (0,77 vs 0,55).

L'algorithme est donc inéquitable.

De plus, il n'est pas fiable pour prédire la récidive dans le groupe B et pour prédire la non-récidive dans le groupe A car dans ces deux cas, il ne fait pas mieux que le hasard (0,50). Prendre de mauvais indicateurs conduit à un **biais statistique**.

Erratum

Activité 2, document 1.

Inverser les logos Linux et Windows. Pour Windows, c'est « Exécutable : .exe, .bin » et pour Linux, c'est « Exécutable : rien, .bin ».

Activité 4, document 3.

Au deuxième paragraphe, il faut écrire : « Par exemple, pour une personne dont le test serait **positif**, [...] ».

Exercice 7.

Dans l'énoncé, changer étoiles bleues en étoiles roses.
Pour la question 2, il faut écrire : « Calculer le **nombre** de faux positifs dans cette situation. »

Exercice 10.

Dans l'énoncé, il faut remplacer la deuxième liste par $L2 = [5, 2, 4, 3, 6, 0]$ et la troisième liste par $L3 = [1, 2, 6, 3, 2, 4]$.

Exercice 14.

Dans l'énoncé, il faut écrire : « [...] tester si un nombre **entier supérieur ou égal à 2** est premier ou non. [...] ».

Exercice 19.

Remplacer le titre par « L'IA, un modèle à évaluer ».
Question 3, il faut écrire : « [...] L'IA s'est donc trompée. **On teste ensuite l'IA avec une autre photo supplémentaire.** Donner le [...] ».

Ces modifications ont été effectuées dans le manuel élève.

