









	<h1>Introduction à l'IA</h1>	
---	------------------------------	---

1 Introduction

L'intelligence artificielle (IA) est « l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine.

Qu'est-ce que l'intelligence ?

- Apprentissage : être capable d'acquérir et de traiter une nouvelle information ;
- Raisonnement : être capable de manipuler l'information de différentes manières ;
- Compréhension : pouvoir apprécier le résultat de la manipulation de l'information ;
- Perception de la vérité : savoir déterminer la validité de l'information manipulée ;
- Vision des liens : percevoir la manière dont les données validées interagissent avec d'autres données ;
- Prise en compte du sens : appliquer des vérités à des situations particulières ;
- Distinction entre les faits et les croyances : pouvoir déterminer si les données sont convenablement étayées par des sources prouvables.

Type d'intelligence	Visuelle	Kinesthésique	Créative	Interpersonnelle	Intrapersonnelle	Linguistique	Logique et mathématique
Exemple	Déplacement d'un robot	Opération chirurgicale	Dessiner en autonomie	Répondre à une question oralement	Résoudre un problème	Analyse d'informations parlées et écrites	Jeu d'échec
Potentiel de simulation	++	++	---	+	---	+	+++
							

L'intelligence suit souvent un processus qu'un système informatique peut imiter.

Algorithme général :

Fixer un objectif

Tant que l'objectif n'est pas atteint et que les possibilités ne sont pas épuisées, faire

Estimer la valeur d'une information connue au regard de l'objectif

Manipuler les données

Définir les valeurs de vérités

Déterminer si l'objectif est atteint

Modifier l'objectif en tenant compte des nouvelles données

Fin tant que

Voir en annexe1 : **Le test de Turing**

2 les applications actuelles de l'IA

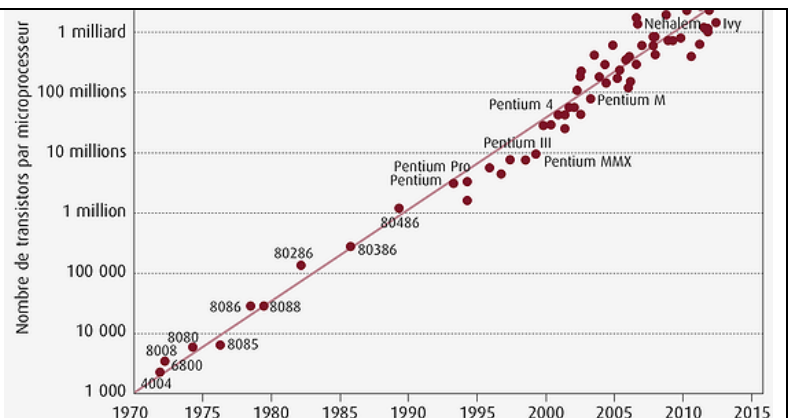
Voir la [vidéo introduction au machine learning](#)

- Détection des fraudes : achat inhabituel sur le net ;
- Besoins médicaux : monitoring, thérapie ;
- Analyse complexe : diagnostic médical, prédiction ;
- Automatisation : gestion de production dans une usine ;
- Service à la clientèle : Aide au dépannage en ligne, diagnostic ;
- Systèmes de sécurité : Prendre le contrôle d'un véhicule en cas de nécessité ;
- Efficacité d'une machine : Optimisation de la consommation grâce à la gestion des ressources ;
- Effectuer des corrections : saisie intuitive sur smartphone ;
- Suggestions : produit recommandé pour vous ;
- Automatiser la traduction : traducteurs en ligne ;
- Créer des liens : réseaux sociaux ;
- Embellir la perception sensorielle humaine : traitement fausses couleurs, réalité augmentée.

3 La loi de Moore

La loi de Moore concerne l'évolution de la puissance de calcul des ordinateurs et de la complexité du matériel informatique.

La première de ces lois est émise par l'ingénieur Gordon E. Moore en 1965, lorsque celui-ci postule sur une poursuite du doublement de la complexité des semi-conducteurs tous les ans à coût constant.



Depuis 1970, les processeurs ne cessent d'évoluer par l'accroissement de la [densité des transistors](#) sur une puce. La puissance d'un microcontrôleur actuel permet facilement d'intégrer des algorithmes d'IA comme le « **machine learning** ».

4 le « big data »



Littéralement, ces termes signifient mégadonnées, grosses données ou encore données massives. Ils désignent un ensemble très volumineux de données qu'aucun outil classique de gestion de base de données ou de gestion de l'information ne peut vraiment travailler. En effet, nous procréons environ 2,5 quintillions (18 zéros) d'octets de données tous les jours. Ce sont les informations provenant de partout : messages que nous nous envoyons, vidéos que nous publions, informations climatiques, signaux GPS, enregistrements transactionnels d'achats en ligne etc. Ces données sont baptisées Big Data ou volumes massifs de données. Les géants du Web, (Facebook et Google), ont été les tous premiers à déployer ce type de technologie.

Le Big Data et l'intelligence artificielle sont deux technologies inextricablement liées, au point que l'on peut parler d'une Big Data Intelligence.

5 L'apprentissage machine (machine learning)

Si l'intelligence artificielle est un concept visant à **simuler** un ou des comportements humains, le « machine learning » n'est qu'une méthode pour atteindre la création d'une intelligence artificielle.

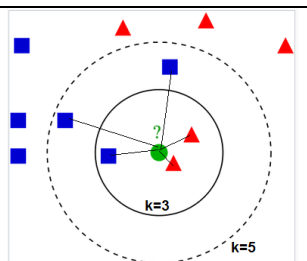
A partir de la vidéo d'introduction au « [machine learning](#) », compléter le tableau ci-dessous en donnant les 3 algorithmes d'apprentissage.

	Algorithme d'apprentissage	Caractéristiques	Applications
1	Apprentissage automatique supervisé	<ul style="list-style-type: none"> - Technique la plus utilisée ; - Modifie son calcul pour minimiser l'erreur ; - Auto évaluation (phase d'entraînement). 	<ul style="list-style-type: none"> - Evaluer un bien ; - Reconnaître une personne ou une forme sur une photo.
2	Apprentissage automatique non supervisé	<ul style="list-style-type: none"> - Pas d'auto évaluation ; - Reconnaît des différences ou des ressemblances. 	<ul style="list-style-type: none"> - Trier des formes géométriques par famille ; - Etudes de marché ; - Recherche scientifique ; - Détection d'anomalies.
3	Apprentissage par renforcement	<ul style="list-style-type: none"> - La machine génère sa propre expérience par l'association état / récompense en fonction d'une action réalisée par elle-même. 	<ul style="list-style-type: none"> - Voiture autonome ; - Jeux vidéo ; - Robotique ; - Prises de décision.

6 Classification et régression

Les problèmes de **Régression** et les problèmes de **Classification** sont utilisés dans les algorithmes de « machine learning ».

6.1 Classification

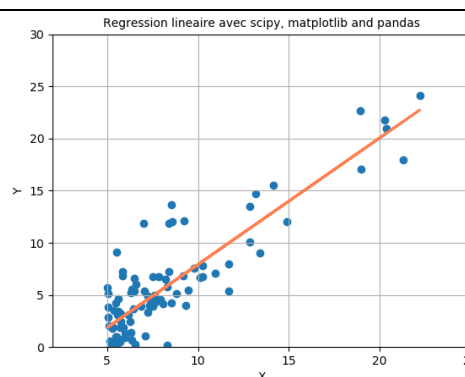


Dans un problème de classification, on cherche à classer un objet dans différentes classes d'appartenance comme nous avons pu le faire en 1ere avec l'algorithme des k plus proches voisins.

Par exemple :

- Prédire si un email est un spam. (*classe* $y = 1$) ou non (*classe* $y = 0$) selon le nombre de liens présents dans l'email (x).

6.2 Régression



Dans les problèmes de régression, on cherche à prédire la valeur d'une variable continue, c'est-à-dire une variable qui peut prendre une infinité de valeurs.

Par exemple :

- Prédire le prix d'un appartement (y) selon sa surface habitable (x).
- Prédire la quantité d'essence consommée (y) selon la distance parcourue (x).

Exemple de régression linéaire

Régression linéaire

```
#https://mrmint.fr/regression-lineaire-python-pratique

import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

#slope : coefficient directeur
#intercept : ordonnée à l'origine

def predict(x):
    return slope * x + intercept

df = pd.read_csv("univariate_linear_regression_dataset.csv")
print ('nombre de valeurs dans le fichier csv',len(df))
X = df.iloc[0:len(df),0] #selection de la première colonne de notre dataset
Y = df.iloc[0:len(df),1]

axes = plt.axes()
axes.set_xlim([2, 25]) #redimensionnement des axes
axes.set_ylim([0, 30])
axes.grid() #affichage d'une grille
plt.scatter(X,Y) #tracé des points

#https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html
#calcul des coefficients
slope, intercept, r_value, p_value, std_err = stats.linregress(X, Y)

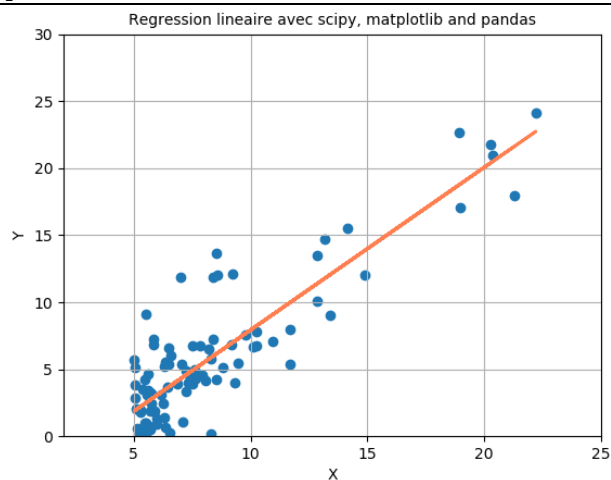
#calcul du tableau de valeurs à l'aide des coefs slope et intercept
droite = predict(X)

plt.plot(X, droite, color='coral', linewidth=2)
plt.title("Regression lineaire avec scipy, matplotlib and pandas",fontsize=10)
plt.xlabel('X')
plt.ylabel('Y')
plt.savefig("regression_lineaire.png")

print('-----')
print('Y=',slope,'* X +',intercept)
valeur=20.27
print('Y =',predict(valeur),' pour X =',valeur)

plt.show()
```

Résultat



```
nombre de valeurs dans le fichier csv 96
-----
Y= 1.213547253908358 * X + -
4.211504005424086
Y = 20.38709883129833 pour X = 20.27
```

7 L'apprentissage automatique supervisé

Voir la Vidéo sur [l'apprentissage automatique supervisé](#)

L'apprentissage automatique, également appelé apprentissage machine ou apprentissage artificiel et en anglais machine learning, est une forme d'intelligence artificielle qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite.

Les 4 notions de l'apprentissage supervisé :

1	DataSet : relation qui relie x feature à y target (la cible)	$\begin{array}{ccc} X & \longrightarrow & Y \\ \text{CHIEN} & \longrightarrow & \text{狗} \end{array}$
2	Modèle mathématique $y=f(x)$	Par exemple $y=ax^2+bx+c$, avec a,b,c les paramètres du modèle.
3	Fonction coût	Ensemble des erreurs par rapport à la dataSet. Avoir un bon modèle, c'est avoir un modèle qui nous donne de petites erreurs, donc une petite fonction coût.
4	Algorithme de minimisation	L'objectif central en Supervised Learning, c'est de trouver les paramètres du modèle qui minimisent la Fonction Coût. Pour cela, on utilise un algorithme d'apprentissage.

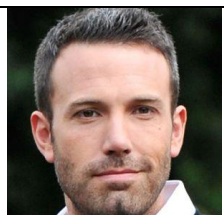
Exemple reconnaissance d'une personne (voir TP OpenCv):

- Phase 1 : Entrainement : apprentissage des visages

A partir des photos suivantes, l'ordinateur va générer un fichier « modèle » au format YAML (Yet Another Markup Language).



- Phase 2 : Reconnaissance d'une nouvelle photographie

	La photo ci-contre n'a pas été apprise. Cependant le programme va bien reconnaître la personne de Ben Affleck à l'aide du fichier modèle généré précédemment.
--	---

L'apprentissage supervisé nécessite la participation et la validation par l'humain puisqu'il faut à l'origine fournir une série de photographies à partir de laquelle la personne va être authentifiée.

8 L'apprentissage automatique non supervisé

L'apprentissage non supervisé est utilisé lorsque le problème nécessite une quantité massive de données non étiquetées, par exemple, les applications de réseaux sociaux, telles que Twitter, Instagram et Snapchat. Pour comprendre le sens de ces données, il est nécessaire d'utiliser des algorithmes qui classifient les données en fonction des tendances.

L'apprentissage non supervisé mène un processus itératif, analysant les données sans intervention humaine. Cette technique peut être utilisée pour la détection de spams envoyés par e-mail.

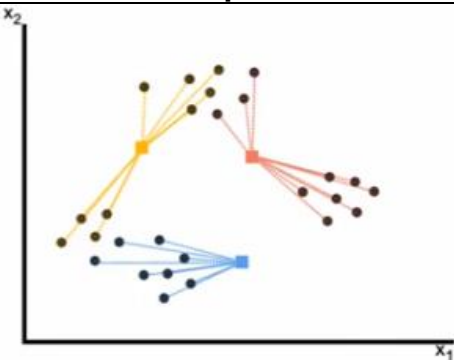
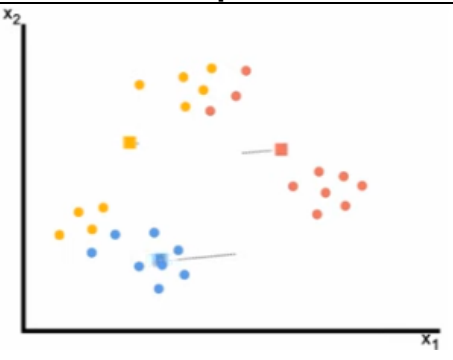
A partir de la vidéo « [apprentissage non supervisé](#) », (de 0' à 3'18'') compléter le tableau ci-dessous en rappelant les principales applications de cette technique d'intelligence artificielle.

Objectif :	Exemples :	Algorithme :
Classer des données en fonction de leur ressemblance	Classer des photos, tweets ; Segmenter la clientèle d'une entreprise.	K-mean clustering
Détection d'anomalies	Systèmes de sécurité ; Détection d'hacking ; Fraudes bancaires ; Défaillances techniques.	Isolation forest
Réduction de dimension	Simplification d'une structure de données tout en conservant les principales informations	L'analyse en composantes principales (PCA)

8.1 Algorithme K-mean clustering

Le K-Mean Clustering est sans doute l'algorithme itératif le plus populaire pour les problèmes de Clustering (regrouper des données selon leur structure commune). L'algorithme fonctionne en 2 étapes répétées en boucle. On commence par placer au hasard un nombre **K** de **centres** dans notre nuage de points. Dans l'exemple ci-dessous, $K=3$.

A partir de la vidéo « [apprentissage non supervisé](#) », (de 3'18'' à 14'53'') compléter le tableau ci-dessous en rappelant les deux étapes de l'algorithme K-mean clustering.

Etape 1	Etape 2
	
Affectation des points au centre le plus proche	Déplacement du centre à la moyenne du cluster

L'algorithme K-mean clustering cherche la position des centres qui minimise la distance entre les points d'un cluster et le centre de ce dernier.

Nom du programme : K-Mean Clustering.py

```
#Algorithme K-mean clustering

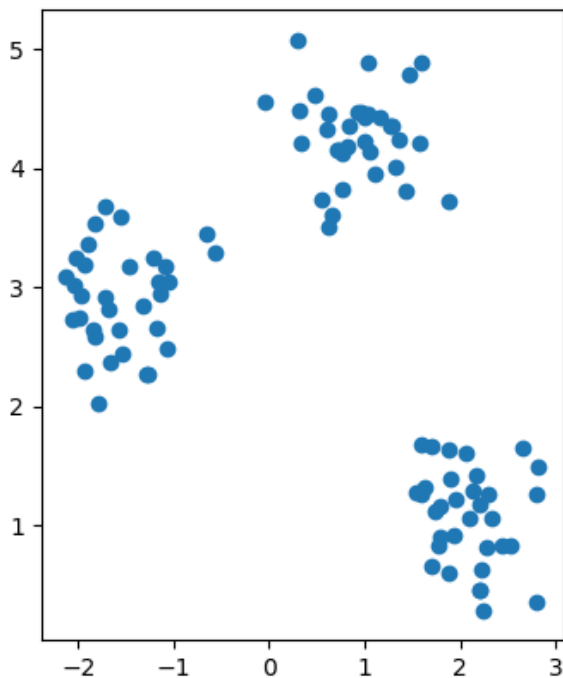
#https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html
#http://python-simple.com/python-matplotlib/scatterplot.php

import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
# Génération de données
X, y = make_blobs(n_samples=100, centers=3, cluster_std=0.4, random_state=0)
fig, ax = plt.subplots(nrows=1, ncols=2)
ax[0].scatter(X[:,0], X[:,1]) #nuage de points de couleur identique

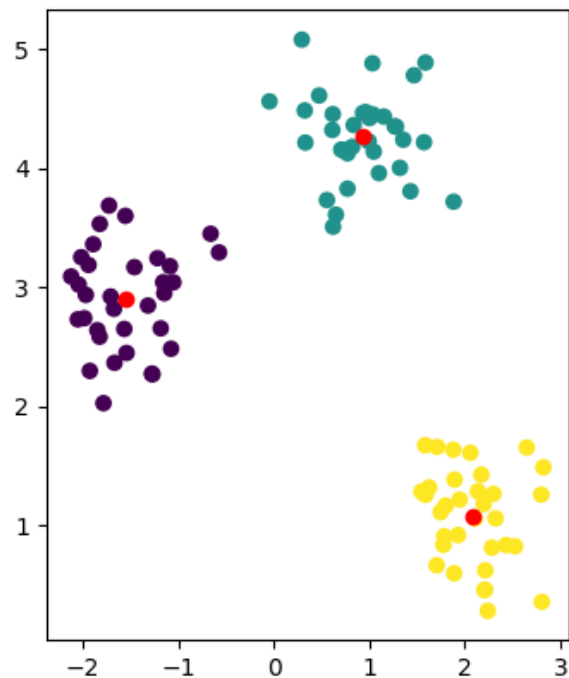
#Algorithme K-mean clustering
model = KMeans(n_clusters=3)
model.fit(X)
model.predict(X)

#Affichage des 3 nuages de points classés
ax[1].scatter(X[:,0], X[:,1], c=model.predict(X))
ax[1].scatter(model.cluster_centers_[0], model.cluster_centers_[0], c='r')
print(model.score(X)) #score de la prédiction
plt.show()
```

Nuage de points avant traitement par l'algorithme K-mean clustering.



Nuage de points après traitement k-mean clustering.



Tester le programme avec d'autres paramètres concernant la génération du nuage de points ($n_samples=300$, $centers=3$, $cluster_std=0.8$, $random_state=2$) en conservant $centers=3$.

8.2 La méthode du coude

Dans l'analyse des clusters, la méthode du coude est utilisée pour déterminer le nombre de clusters dans un ensemble de données.

Nom du programme : Elbow method.py

```
#Elbow method
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
# Génération de données
X, y = make_blobs(n_samples=100, centers=3, cluster_std=0.8, random_state=0)
fig, ax = plt.subplots(nrows=1, ncols=2)
ax[0].scatter(X[:,0], X[:,1]) #nuage de points de couleur identique

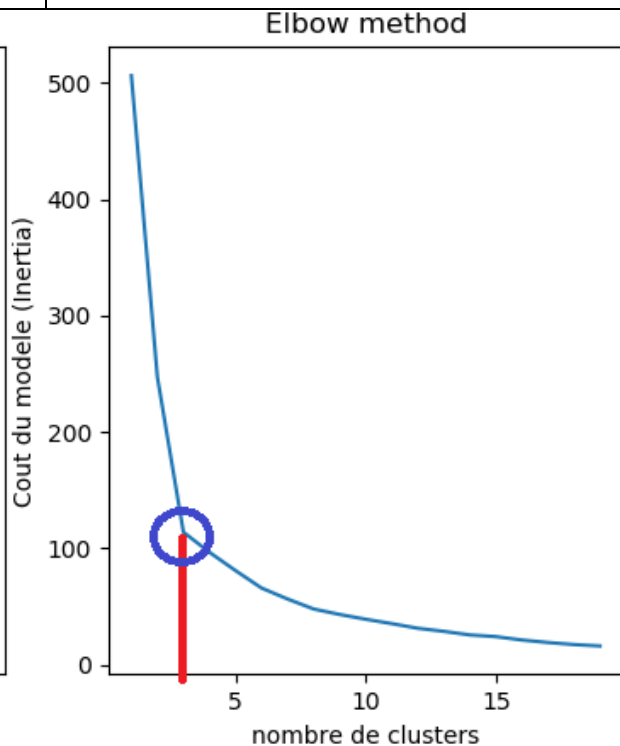
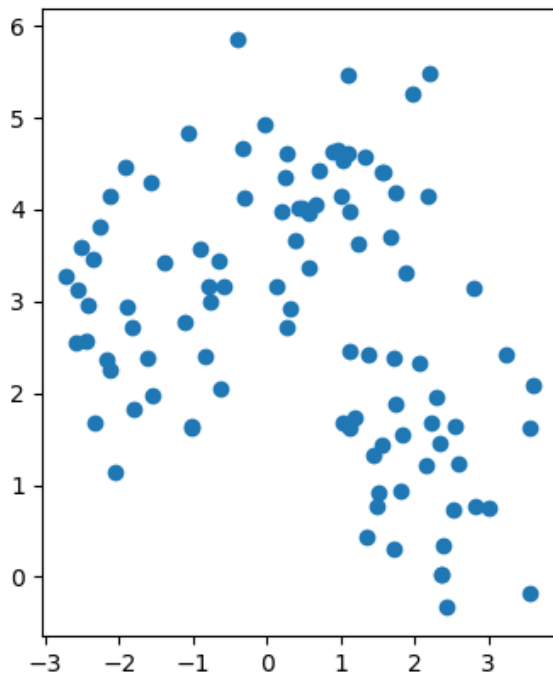
inertia = []
K_range = range(1, 20)
for k in K_range:
    model = KMeans(n_clusters=k).fit(X)
    inertia.append(model.inertia_)

ax[1].plot(K_range, inertia)
ax[1].set(xlabel='nombre de clusters', ylabel='Cout du modele (Inertia)', title='Elbow method')

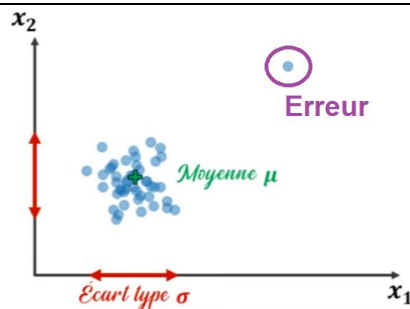
plt.show()
```

Nuage de points quelconque

Le repérage du coude (elbow) donne la meilleure valeur de K (nombre de clusters)



8.3 Détection d'anomalies



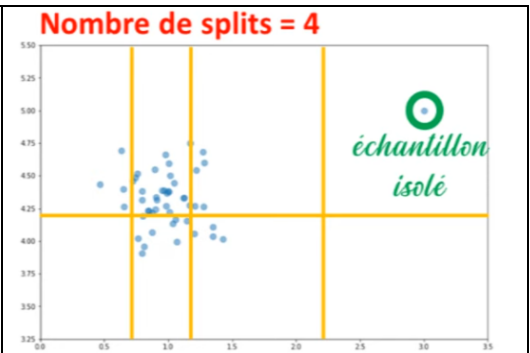
Il est fréquent qu'un Dataset (Catalogue des données) contienne quelques anomalies, voire des erreurs, qu'il faut supprimer pour ne pas biaiser l'apprentissage de la machine (il ne faut pas que la machine apprenne quelque chose de faux).

L'algorithme « isolation forest » permet de détecter des anomalies dans un jeu de données. Il isole les données atypiques, autrement dit celles qui sont trop différentes de la plupart des autres données.

Algorithme « isolation forest ». Voir vidéo « [apprentissage non supervisé](#) », (de 14'53" à 21'40")

- On effectue une série de découpes(split) aléatoires.
- On compte le nombre de découpes qu'il faut faire pour pouvoir isoler les échantillons.

Si le nombre de découpes est faible, alors il y a une forte probabilité qu'il y ait une anomalie.



Nom du programme : isolation forest.py

```
# Algorithme isolation forest

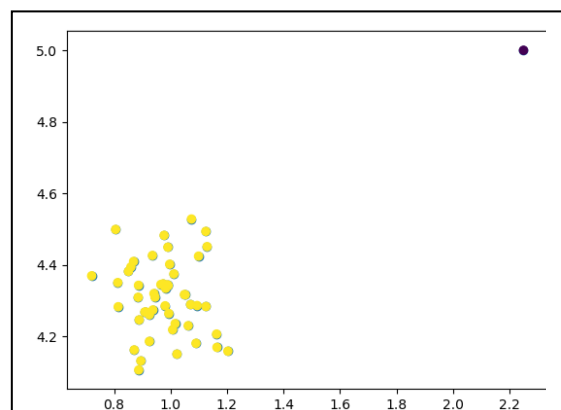
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.ensemble import IsolationForest

#creation du nuage de points
X, y = make_blobs(n_samples=50, centers=1, cluster_std=0.1, random_state=0)
#ajout de l'erreur
X[-1,:] = np.array([2.25, 5])

plt.scatter(X[:,0], X[:, 1])
#plt.show()

#taux de contamination estimé ici 0.01=1%
model = IsolationForest(contamination=0.01)
model.fit(X)

plt.scatter(X[:,0], X[:, 1], c=model.predict(X))
plt.show()
```



Regarder la vidéo « [apprentissage non supervisé](#) », (de 21'40" à 27'00"), puis tester l'exemple proposé pour nettoyer la dataset image composée de chiffres manuscrits.

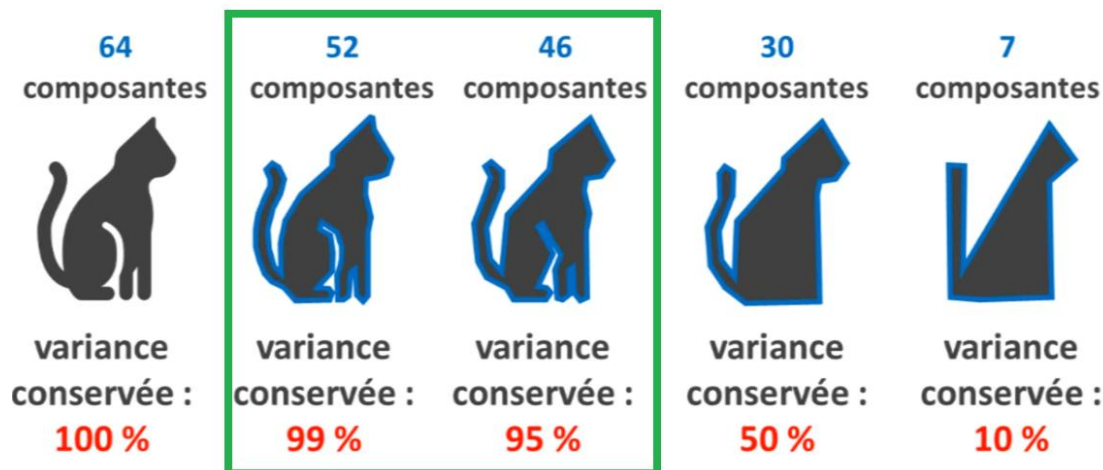
Ne pas oublier l'instruction `plt.show()` pour afficher l'image dans edupython.



8.4 Réduction de dimension (PCA : Principal Component Analysis)

Le principe est de réduire la complexité superflue d'un dataset en projetant ses données dans un espace de plus petite dimension. De ce fait le nombre de variables en entrées diminue et cela permet d'accélérer l'apprentissage.

Plus on réduit la dimension d'un dataset, plus on réduit en qualité, c'est-à-dire en variance. Le but est de réduire au maximum la dimension d'un dataset, tout en minimisant la perte de qualité. Il est conseillé de garder 95 à 99% de la variance originelle des données.



Regarder la vidéo « [apprentissage non supervisé](#) », (de 27'00" à 40'00"), puis tester l'exemple proposé pour nettoyer la dataset image composée de chiffres manuscrits.

8.4.1 Analyse des données

Nom du programme : PCA analyse.py

```
# Algorithme PCA analyse

import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.datasets import load_digits

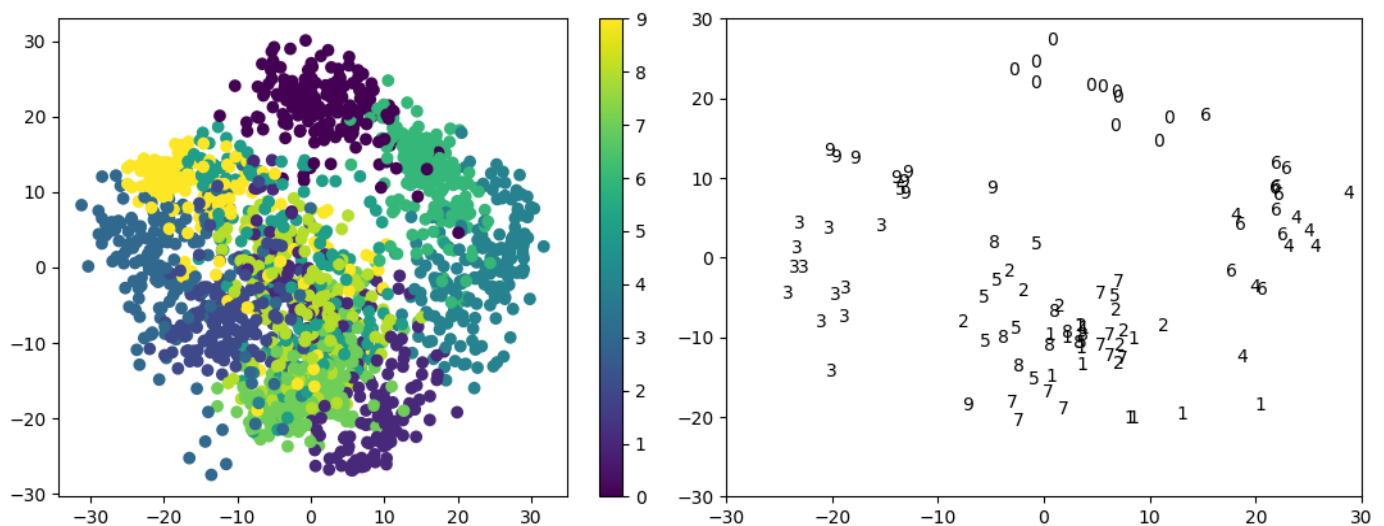
digits = load_digits()
images = digits.images
X = digits.data
y = digits.target

model = PCA(n_components=2)
model.fit(X)
x_pca = model.transform(X)
plt.scatter(x_pca[:,0], x_pca[:,1], c=y)
plt.colorbar()

plt.figure()
plt.xlim(-30, 30)
plt.ylim(-30, 30)

for i in range(100):
    plt.text(x_pca[i,0], x_pca[i,1], str(y[i]))
plt.show()
```

Résultat dans la console :



On remarque que les zones de couleurs correspondent aux différents chiffres. Par exemple le chiffre 9 correspond aux points jaunes. On peut également représenter le même graphique de manière plus lisible à droite.

8.4.2 Compression des données

Nom du programme : **PCA compression.py**

```
# Algorithme PCA compression

import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.datasets import load_digits

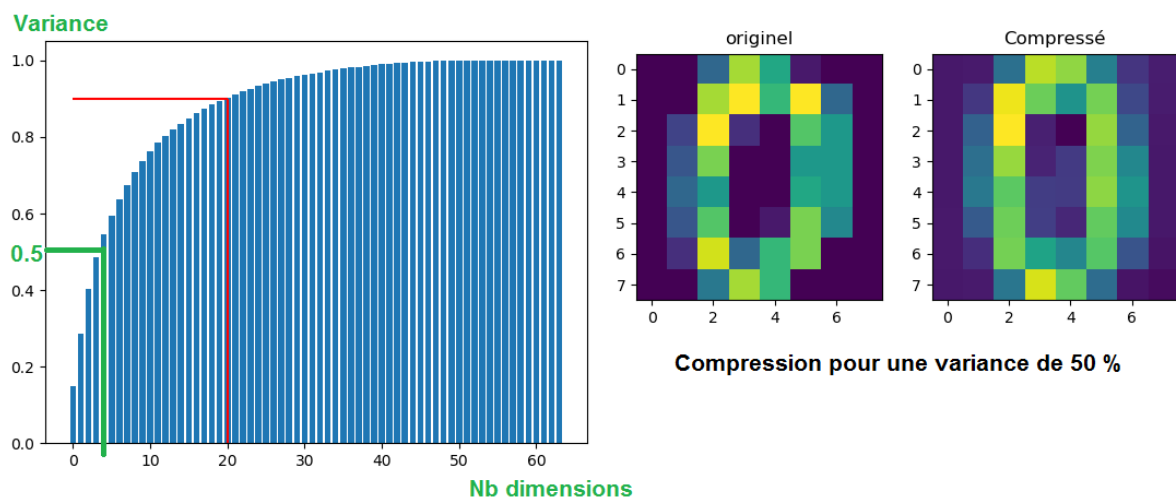
digits = load_digits()
images = digits.images
X = digits.data
print(X.shape)

n_dims = X.shape[1]
model = PCA(n_components=n_dims)
model.fit(X)
variances = model.explained_variance_ratio_
meilleur_dims = np.argmax(np.cumsum(variances) > 0.90)

#historique en barre
plt.bar(range(n_dims), np.cumsum(variances))
plt.hlines(0.90, 0, meilleur_dims, colors='r')
plt.vlines(meilleur_dims, 0, 0.90, colors='r')
plt.show()

#compression avec une variance de 50%
model = PCA(n_components=0.5)
X_reduced = model.fit_transform(X)
X_recovered=model.inverse_transform(X_reduced)
plt.subplot(1, 2, 1)
plt.imshow(X[0,:].reshape((8,8)))
plt.title('originel')
plt.subplot(1, 2, 2)
plt.imshow(X_recovered[0].reshape((8,8)))
plt.title('Compressé')
print(model.n_components_)
plt.show()
```

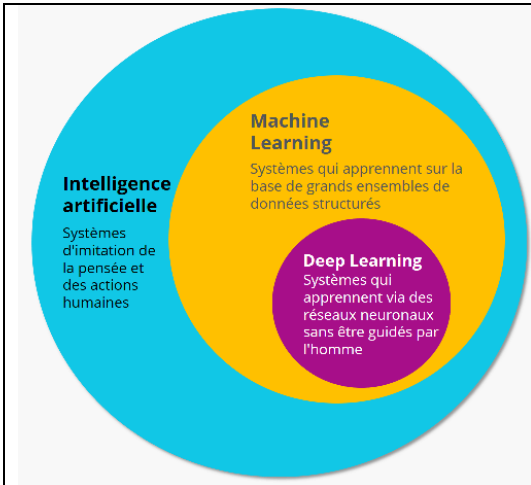
Résultat dans la console :



Tester le programme précédent en faisant varier le pourcentage de variance.

9 Le deep learning (apprentissage en profondeur) et les réseaux de neurones.

Regarder la vidéo d'introduction sur le [deep learning](#) de la chaîne YT science étonnante.



Le Deep learning ou apprentissage profond est l'une des technologies principales du Machine learning. Avec le Deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificiels. Les réseaux sont composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

Les réseaux neuronaux et l'apprentissage en profondeur sont souvent utilisés dans les applications de reconnaissance d'image, de communication orale et de vision numérique.

Les réseaux de neurones artificiels, inspirés du comportement du cerveau humain, permettent de créer de l'intelligence artificielle. Ils sont utilisés principalement à travers l'apprentissage non supervisé, ils servent à prédire, à identifier et à classer les données.

Le concept des réseaux de neurones artificiels fut inventé en 1943 par deux chercheurs de l'Université de Chicago : le neurophysicien Warren McCulloch, et le mathématicien Walter Pitts.

En 1957, le Perceptron fut inventé. Il s'agit du plus ancien algorithme de Machine Learning, conçu pour effectuer des tâches de reconnaissance de patterns complexes. C'est cet algorithme qui permettra plus tard aux machines d'apprendre à reconnaître des objets sur des images.

Malheureusement, à l'époque, les réseaux de neurones étaient limités par les ressources techniques. Par exemple, les ordinateurs n'étaient pas assez puissants pour traiter les données nécessaires au fonctionnement des réseaux de neurones. C'est la raison pour laquelle la recherche dans le domaine des Neural Networks est restée en sommeil durant de longues années.

Exemple de réseau de neurones composés de plusieurs couches

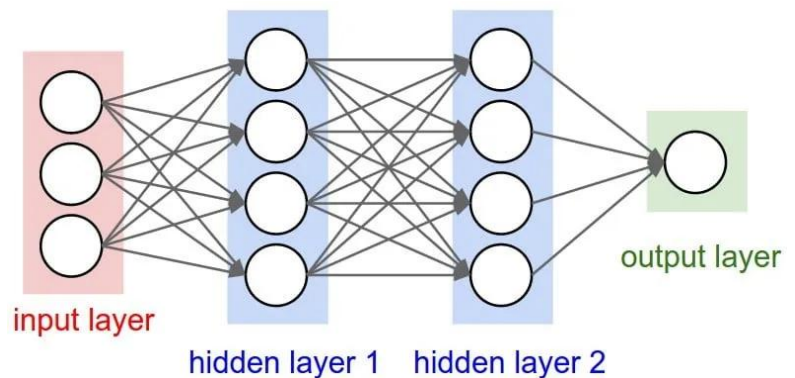
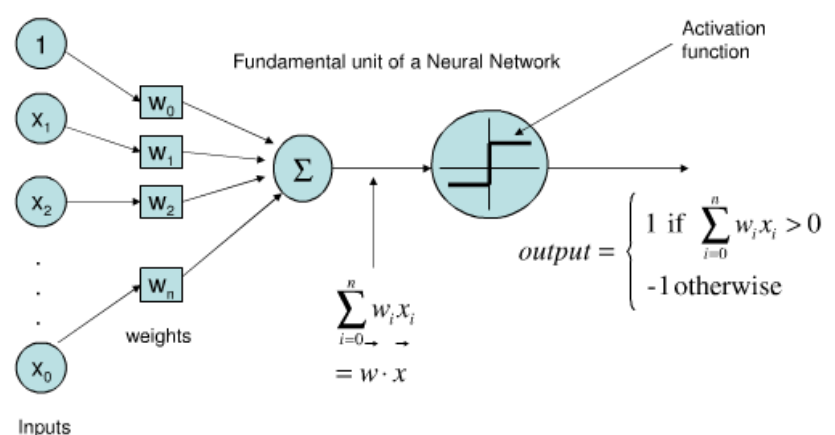


Diagramme d'un Perceptron



Il aura fallu attendre le début des années 2010, avec l'essor du Big Data pour que les Data Scientists disposent des données et de la puissance de calcul nécessaires pour exécuter des réseaux de neurones complexes.

Regarder la vidéo : « [Comprendre le DeepLearning et les Réseaux de neurones en 10 mins](#) » ou en anglais « [Mais *qu'est-ce* qu'un réseau de neurones ?](#) ».

9 L'algorithme descente de gradient (Gradient descent)

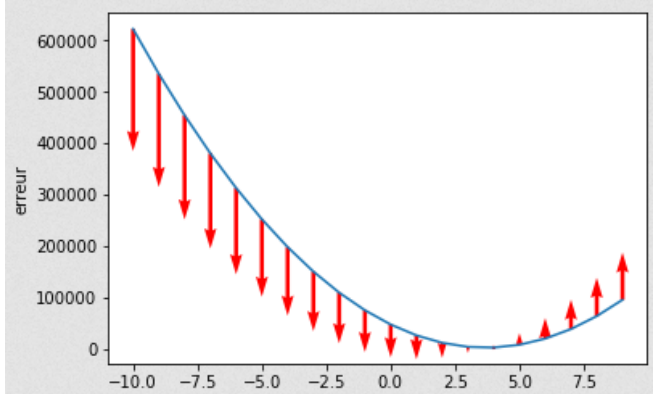
L'algorithme descente de gradient permet de trouver rapidement le minimum d'une fonction mathématique. Pour faire simple, trouver x tel que $f(x)$ soit le plus petit possible. Cette méthode est très utilisée en IA avec les réseaux de neurones artificiels

Regarder la vidéo « [descente de gradient](#) »

Comment trouver le point le plus bas dans une vallée ?



- Calcul de la pente (dérivée)
- Petit pas dans la descente et ainsi de suite



```
#descente de gradient
#https://docs.scipy.org/doc/scipy/refere
#nce/generated/scipy.misc.derivative.html
```

```
from scipy import misc
import matplotlib.pyplot as plt
import numpy as np
```

```
# Function Definition
```

```
def fonction(x):
    return 3*x*x+2*x+1
```

```
# Plot function
x = np.arange(-2.0, 2.0, 0.01)
y = fonction(x)
plt.plot(x, y, 'r-')
```

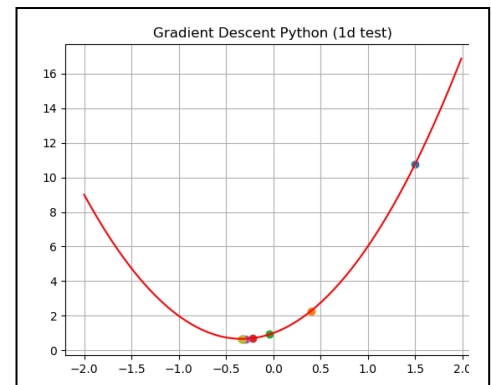
```
# Gradient Descent
```

```
alpha = 0.1 # learning rate
nb_max_iter = 100 # Nb max d'iteration
eps = 0.0001 # stop condition
```

```
x0 = 1.5 # start point
y0 = fonction(x0)
plt.scatter(x0, fonction(x0))
```

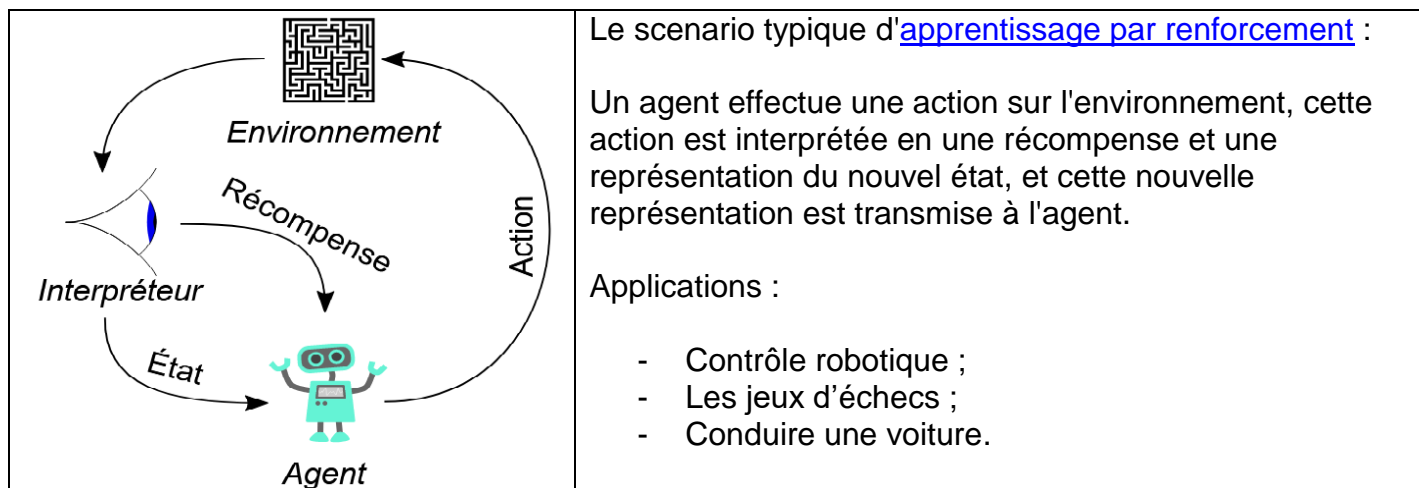
```
cond = eps + 10.0 # start with cond greater than eps (assumption)
nb_iter = 0
tmp_y = y0
while cond > eps and nb_iter < nb_max_iter:
    x0 = x0 - alpha * misc.derivative(fonction, x0)
    y0 = fonction(x0)
    nb_iter = nb_iter + 1
    cond = abs( tmp_y - y0 )
    tmp_y = y0
    print (x0,y0,cond)
    plt.scatter(x0, y0)
```

```
plt.title("Gradient Descent
Python (1d test)")
plt.grid()
plt.show()
```



10 Apprentissage par renforcement

L'apprentissage par renforcement est un modèle d'apprentissage comportemental. L'algorithme reçoit un feedback de l'analyse des données et guide l'utilisateur vers le meilleur résultat. L'apprentissage par renforcement diffère des autres types d'apprentissage supervisé, car le système n'est pas formé avec un ensemble de données exemple. Au lieu de cela, le système apprend plutôt par le biais d'une méthode d'essais et d'erreurs. Par conséquent, une séquence de décisions fructueuses aboutit au renforcement du processus, car c'est lui qui résout le plus efficacement le problème posé.



Regarder la vidéo « [Apprentissage par renforcement](#) »

11 Conclusion

Après avoir progressivement envahi notre quotidien avec des voitures autonomes, des objets connectés et des assistants intelligents tels que Siri, Alexa ou « Ok Google », l'intelligence artificielle rend les machines capables de nous transporter d'un point A à un point B.

Certains craignent que les machines aient un poids trop important en ce qui concerne la vie humaine. L'une des grandes interrogations est donc de se demander où s'arrêtera ce progrès et si les machines vont nous renverser comme dans les pires scénarios hollywoodiens. On peut se rassurer, on en est très loin... !

Il est peu probable qu'une machine parvienne à simuler des sentiments propres à l'humain comme l'amour, la compassion ou la tristesse. L'intelligence artificielle excelle dans des domaines très spécifiques, dans des tâches répétitives et encadrées mais ne peut pas prendre d'initiative.

IA peut-elle véritablement être considérée comme un atout pour la transition écologique malgré son impact environnemental (consommatrice d'énergie) plutôt négatif ? L'avenir nous le dira...

Sources :

<https://www.youtube.com/c/MachineLearnia/videos>
<https://www.artificiel.net/test-de-turing>
<https://www.ibm.com/fr-fr/analytics/machine-learning>
<https://www.franceculture.fr/litterature/quand-lecrivain-de-science-fiction-isaac-asimov-predisait-le-futur>
<https://openclassrooms.com/fr/courses/4011851-initiez-vous-au-machine-learning/4020611-identifiez-les-differents-types-dapprentissage-automatiques>

<https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition>
<https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>
<https://datascientest.com/intelligence-artificielle-quelles-limites>

Annexe 1

Le test de Turing

[Alan Turing](#), né le 23 juin 1912 à Londres et mort le 7 juin 1954 à Wilmslow, est un mathématicien et cryptologue britannique, auteur de travaux qui fondent scientifiquement l'informatique.

Le test de Turing est un test permettant de vérifier la capacité d'une machine à faire preuve de signes d'intelligence humaine. Encore aujourd'hui, ce test fait figure de standard pour déterminer l'intelligence d'une machine, en dépit de nombreuses critiques formulées au fil des années.

Le principe du Test de Turing est simple. Un évaluateur humain est chargé de juger une conversation textuelle entre un humain et une machine. L'évaluateur sait que l'un des deux participants est une machine, mais ne sait pas lequel. S'il n'est pas en mesure de discerner l'homme de la machine après 5 minutes de conversation, la machine a passé le test avec succès. Le test ne mesure pas la capacité d'une machine à répondre correctement à une question, mais à quel point ses réponses ressemblent à celles que fournirait un humain.

En 1966, Joseph Weizenbaum a créé un programme capable de passer le test de Turing. Baptisé ELIZA, ce programme était en mesure d'examiner un texte à la recherche de mots-clés pour formuler une réponse cohérente.

```

=====
EEEEEEEE L      IIIIIII ZZZZZZZ      AAA
E         L      I      Z      A  A
E         L      I      Z      A  A
EEEEEE   L      I      Z      A  A
E         L      I      Z      AAAAAA
E         L      I      Z      A  A
EEEEEEEE LLLLLLL IIIIIII ZZZZZZ      A  A
=====
ELIZA > (PLEASE TYPE IN ALL CAPS) WHAT'S YOUR NAME DEAR ?
AMIT
ELIZA > HI, I'M ELIZA. WHAT DO YOU WANT TO TALK ABOUT ?
AMIT > CAN I TALK ABOUT MY PROBLEM ?
ELIZA > SURE... !
AMIT > DUE TO ON GOING EXAMS IN COLLEGE, I HAVE A LOT OF STRESS.
ELIZA > PLEASE RELAX A LITTLE. SLEEP WELL.
AMIT > THANKS FOR YOUR ADVICE.
ELIZA > NO MENTION.
AMIT > BYE
ELIZA > BYE AND KEEP IN TOUCH...
=====

```



Le prix Loebner est une compétition annuelle qui couronne les dialogueurs satisfaisant le mieux les critères du test de Turing (avec lesquels il est le plus difficile de déterminer s'il s'agit d'un robot ou d'un humain).

Chaque année, une médaille de bronze est décernée au programme informatique démontrant le comportement conversationnel le plus proche d'un humain. En revanche, le prix d'argent, basé uniquement sur du texte, et le prix d'or, basé sur le visuel et l'audio, n'ont jamais été remportés.

Une machine, sera-elle un jour capable :

- De penser comme un humain ?
- D'agir rationnellement ?
- D'avoir une conscience ?

Annexe 2

Des robots à l'intelligence artificielle, de l'ordinateur à Internet, [Isaac Asimov](#) écrivain de science-fiction a laissé derrière lui une œuvre visionnaire, ayant anticipé, 50 ans avant tout le monde, nombre des évolutions technologiques de notre temps :

Prédictions d'Isaac Asimov :

- L'avènement de l'ordinateur et d'internet ;
- La visioconférence ;
- Une constellation de satellites rendra possible les appels directs vers n'importe quel point de la terre ;
- L'objet mobile informatisé va "pénétrer dans la maison (IOT) ;
- Inventeur du terme "robotique" ;
- Centrales d'énergie solaire.



[Les Trois lois de la robotique](#), formulées en 1942

- 1 Un robot ne peut porter atteinte à un être humain ni, restant passif, laisser cet être humain exposé au danger ;
- 2 Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres entrent en contradiction avec la première loi ;
- 3 Un robot doit protéger son existence dans la mesure où cette protection n'entre pas en contradiction avec la première ou la deuxième loi.

« Une fois que nous aurons un ordinateur dans chaque maison, chacun d'entre eux connecté à d'énormes bibliothèques, où n'importe qui peut poser n'importe quelle question, et se voir donner une réponse, une référence [...], tout le monde appréciera d'apprendre. »

En 1964, Asimov assurait que si la population continuait à grandir sans être régulée, la société finirait par s'effondrer.

« Il est important que le monde soit ensemble et suffisamment unifié pour faire face aux problèmes qui s'attaquent à notre unité même. ».

A voir la conférence de Jean-Marc Jancovici enregistré à Sablé sur Sarthe :

<https://youtu.be/3vg9kffxPtE>