République Tunisienne Ministère de l’Enseignement Supérieur

et de la Recherche Scientifique

Faculté des Sciences Mathématiques Physiques et Naturelles

de Tunis

DEPARTEMENT DES SCIENCES DE L’INFORMATIQUE

**Cycle d’ingénieur en Data science**

**Rapport de projet**

**Réalisé par :**

***Eya Besbes***

***Et***

***Mayssa Hemdana***

***Réalisation d’un classifieur à base de réseau de neurones (MLPClassifier de Sklearn)***

**Année Universitaire : 2022-2023**

**Table des matières**

Table des matières

[Introduction générale 6](#_Toc132218777)

[3.4 Présentation du projet 8](#_Toc132218778)

[1.1 Contexte du projet : 8](#_Toc132218779)

[1.2 Présentation du projet : 8](#_Toc132218780)

[1.3 Travail demandé : 8](#_Toc132218781)

[I. Présentation des méthodes utilisées et des concepts liés : 10](#_Toc132218782)

[1.1 Définition : 11](#_Toc132218784)

[1.2 Types d’algorithmes du Machine Learning : 11](#_Toc132218785)

[3.1 Définition : 12](#_Toc132218788)

[3.2 Définition du Réseau des neurones : 12](#_Toc132218789)

[3.3 Fonctionnement du réseau des neurones : 13](#_Toc132218795)

[II. Présentation des technologies utilisées dans ce projet : 17](#_Toc132218796)

[III. Présentation des jeux de données utilisés : 18](#_Toc132218797)

[IV. Présentation du jeu de données digits : 23](#_Toc132218799)

[I. Présentation des résultats obtenus en utilisant le jeu de données digits : 29](#_Toc132218801)

[1. Phase d’apprentissage : 29](#_Toc132218802)

[2. Prédiction : 30](#_Toc132218805)

[Dans cette phase nous allons prédire y\_pred à partir des données de x\_test. 30](#_Toc132218806)

[3. Test et évaluation du modèle : 30](#_Toc132218808)

[4.3 Score de l’apprentissage et de test : 30](#_Toc132218809)

[4.4 Affichage des poids et des biais : 30](#_Toc132218813)

[(A) Affichage des poids utilisés entre la 1ère et la 2ème couche : 30](#_Toc132218814)

[(B) Affichage du poids w0 : 31](#_Toc132218816)

[(C) Affichage des biais : 31](#_Toc132218818)

[4.5 Affichage du meilleur score obtenu pendant l'apprentissage: 31](#_Toc132218820)

[4.6 Calcul de la précision, de la justesse, du rappel ainsi que la matrice de confusion: 32](#_Toc132218822)

[(A) Définition de la précision : 32](#_Toc132218823)

[(B) Définition de la justesse : 32](#_Toc132218826)

[(C) Définition du rappel: 32](#_Toc132218828)

[(D) Définition d’une matrice de confusion : 33](#_Toc132218829)

[(E) Résultats de calcul : 33](#_Toc132218831)

[4.7 Affichage de la courbe loss\_curve: 34](#_Toc132218835)

[4.8 Affichage de la courbe learning\_curve : 35](#_Toc132218837)

[4.9 Courbe de validation : 35](#_Toc132218839)

[4.10 Rapport de classification : 36](#_Toc132218841)

[4. Optimisation de MLPClassifier par Grid search : 37](#_Toc132218848)

[4.1 Apprentissage : 37](#_Toc132218849)

[4.2 Test et évaluation : 37](#_Toc132218851)

[(A) Affichage des meilleurs parmaètres : 37](#_Toc132218852)

[(B) Affichage de la moyenne, l’ecart-type, l’alpha, le nombre de couches et le learning\_rate dans chaque itération : 37](#_Toc132218854)

[(C) Rapport de classification : 38](#_Toc132218856)

[(D) La justesse du paramètre learning\_rate\_init pendant chaque itération : 38](#_Toc132218858)

[II. Présentation des résultats obtenus en utilisant le jeu de données Iris : 39](#_Toc132218860)

[1. Apprentissage : 39](#_Toc132218861)

[2. Prédiction : 39](#_Toc132218863)

[3. Test et évaluation du modèle : 39](#_Toc132218865)

[3.1 Affichage du score, de la précision, de la justesse et de la matrice de confusion : 39](#_Toc132218866)

[4. Affichage des poids et des biais utilisés dans l’apprentissage : 41](#_Toc132218872)

[(A) Affichage des poids : 41](#_Toc132218873)

[(B) Affichage des biais : 41](#_Toc132218875)

[5. Traçage des courbes : 41](#_Toc132218877)

[(A) Loss\_curve : 41](#_Toc132218878)

[(B) Learning\_curve : 42](#_Toc132218880)

[(C) Validation\_curve : 42](#_Toc132218882)

[III. Etude comparative entre les différents classifieurs sur le même jeu de données : 43](#_Toc132218884)

[IV. Etude comparative entre Iris et digits: 44](#_Toc132218886)

[Conclusion : 46](#_Toc132218887)

**INTRODUCTION GENERALE**

# Introduction générale

Il s’avère que l’intelligence artificielle joue un rôle majeur au sein de notre société grâce au progrès révolutionnaire qu’elle avait connu à l’ère du numérique. Depuis son émergence en 1950, nous avons assisté à une croissance exponentielle de son potentiel. Ainsi, est-elle devenue omniprésente dans plusieurs domaines tels que le transport, le commerce, la finance, la sécurité etc. Elle avait également fait son chemin dans le domaine de Deep Learning.

C’est ainsi que nous nous sommes intéressés à l’IA et plus particulièrement au Deep Learning. Néanmoins, nous nous limiterons dans ce projet à l’utilisation d’un classifieur à base de réseau de neurones ( MLPClassifier de Sklearn).

De ce fait, nous avons élaboré ce rapport contenant toutes les informations relatives à ce projet qui sera réparti en trois chapitres définis comme suit :

Le premier chapitre sera consacré à la présentation de l’organisme d’accueil et à la description du contexte général du projet.

Le second chapitre, abordera les différentes solutions préexistantes adoptées à ce genre de projet avec une explication de la méthodologie employée.

Finalement, le troisième chapitre portera sur les différents résultats trouvés lors de l’utilisation de nos propres solutions.

**Chapitre I :**

**CONTEXTE GENERAL DU PROJET**

**Introduction :**

Dans ce chapitre nous allons mettre le projet dans son cadre général.

# Présentation du projet

## sprésentation du projet :

Ce projet permet de réaliser un classifieur à base de réseau de neurones. Ce classifieur est un classifier MLPCLassifier de la bibliothèque sklearn, qui permet de classifier un dataset donné en input tout en utilisant les techniques de Deep Learning.

## Travail demandé :

Notre projet consiste à:

* + 1. Importer des jeux de données et les explorer
    2. Diviser les données en données de test et d’apprentissage
    3. Entrainer le jeu de données avec le modèle MLPClassifier de Sklearn
    4. Prédire quelques valeurs avec les données de test
    5. Evaluer le modèle
    6. Comparer le modèle MLPClassifier avec d’autres modèles

**Chapitre II :**

**Présentation et exploration des jeux de données utilisés :**

**Introduction :**

Dans ce chapitre, nous allons décrire dans un premier temps les méthodes utilisées ainsi que les concepts liés à ce projet et puis dans un second temps nous montrerons les différentes technologies qui nous ont permis de réaliser ce projet. Finalement, nous présenterons les jeux de données utilisés.

# Présentation des méthodes utilisées et des concepts liés :

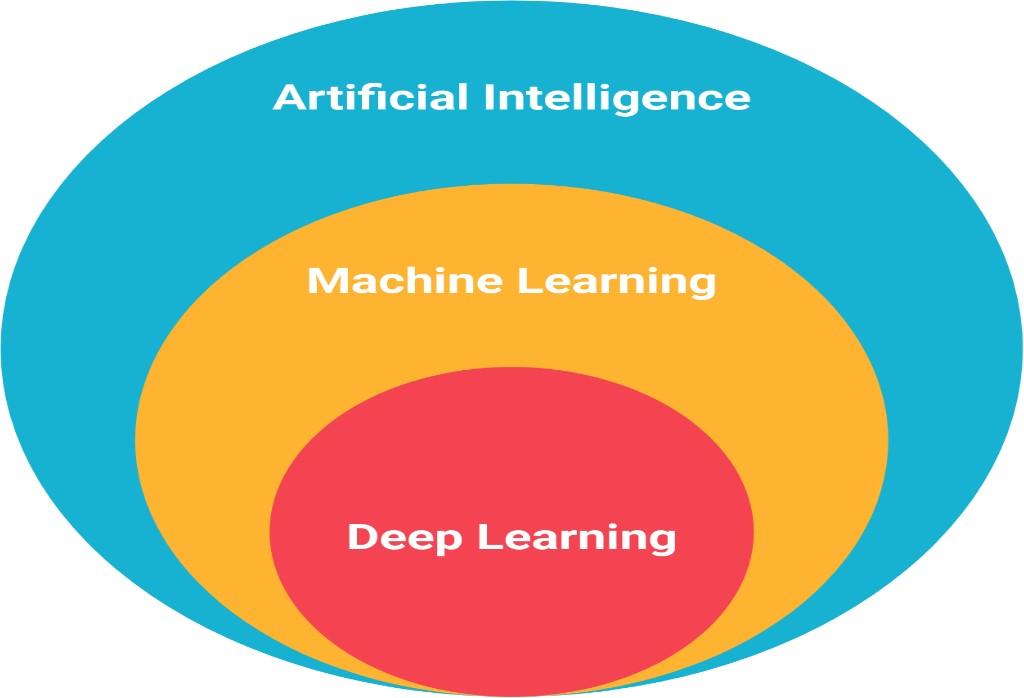
## Dans ce projet nous avons utilisé MLPclassifier de la bibliothèque sklearn. Ce classifieur s’instaure dans le cadre du Deep Learning. Cet ainsi, que nous expliquerons dans cette section ce que c’est que le Deep Learning mais avant de procéder à cette explication il faut tout d’abord comprendre ce que c’est que l’intelligence artificielle, le machine Learning et le réseau de neurones.

* 1. **L’intelligence artificielle :**

L’intelligence artificielle, désigne les systèmes ou les machines qui imitent l'intelligence humaine pour effectuer des tâches et qui peuvent s'améliorer de manière itérative en fonction des informations qu'ils recueillent.

Elle englobe également les sous-domaines de l'apprentissage automatique et de l'apprentissage en profondeur. Ces disciplines sont composées d'algorithmes d'IA qui cherchent à créer des systèmes experts qui exécutent des prévisions ou des classifications sur la base de données d'entrée.

La figure ci-dessous montre la relation entre l’IA, le Machine Learning et le Deep Learning.



* 1. **Le Machine Learning :**

## Définition :

L’apprentissage automatique est une branche de l’intelligence artificielle et une composante importante de la science des données. Il se concentre sur l’utilisation des données et des algorithmes de l’IA en se basant sur des méthodes statistiques permettant de faire des classifications ou des prédictions et de découvrir des informations clés dans les projets d’exploration de données

## Types d’algorithmes du Machine Learning :

Il existe plusieurs algorithmes de Machine Learning qui varient selon le type d’apprentissage utilisé. On en distingue quatre types :

* + - L’apprentissage supervisé : fait des prédictions basées sur un ensemble d’exemples étiquetés que vous fournissez.
    - L’apprentissage non supervisé : utilise des points de données qui ne sont pas étiquetés : l’algorithme les étiquette pour vous en organisant les données ou en décrivant leur structure
    - L’apprentissage semi-supervisé : Il utilise un ensemble de données étiquetées et non-étiquetées. Il se situe entre l’apprentissage supervisé et l’apprentissage non supervisé.
    - L’apprentissage par renforcement : utilise des algorithmes qui apprennent des résultats et décident de l’action à entreprendre ensuite.
  1. **Le Deep Learning :**



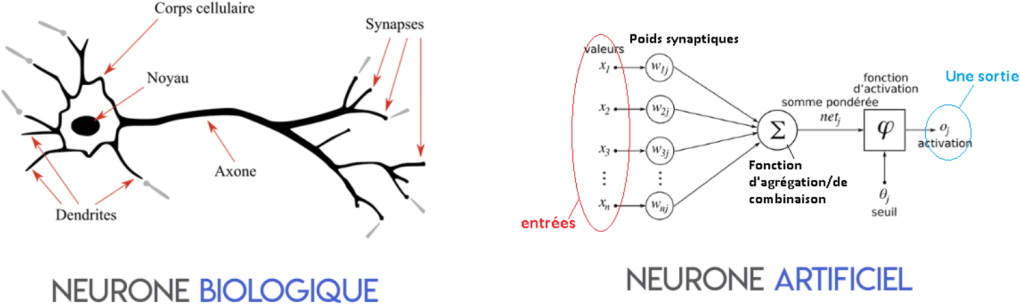
## Définition :

L’apprentissage profond ou encore « Le Deep Learning » est une branche du Machine Learning qui repose sur l’utilisation des réseaux neuronaux artificiels permettant ainsi aux systèmes numériques de comprendre des données non structurées et non étiquetées avec précision tout en les analysant à un haut niveau d'abstraction grâce à une compréhension non linéaire.

## Définition du Réseau des neurones :

Afin de mieux comprendre ce que c’est que le Deep Learning, il faudrait tout d’abord comprendre le principe des réseaux de neurones.

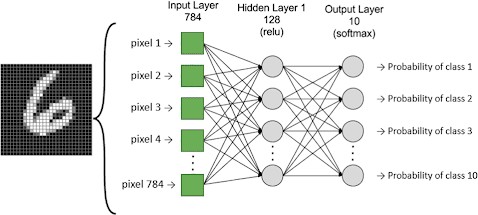
Un réseau de neurones artificiels, est un système informatique matériel et / ou logiciel **dont le fonctionnement est calqué sur celui des neurones du cerveau humain**. Il existe plusieurs types de réseaux de neurones : on en distingue le CNN (Le réseau des neurones de convolution), le RNN : (Le réseau de neurones récurent) ainsi que Le GAN : (réseau de neurones antagoniste génératif) ...





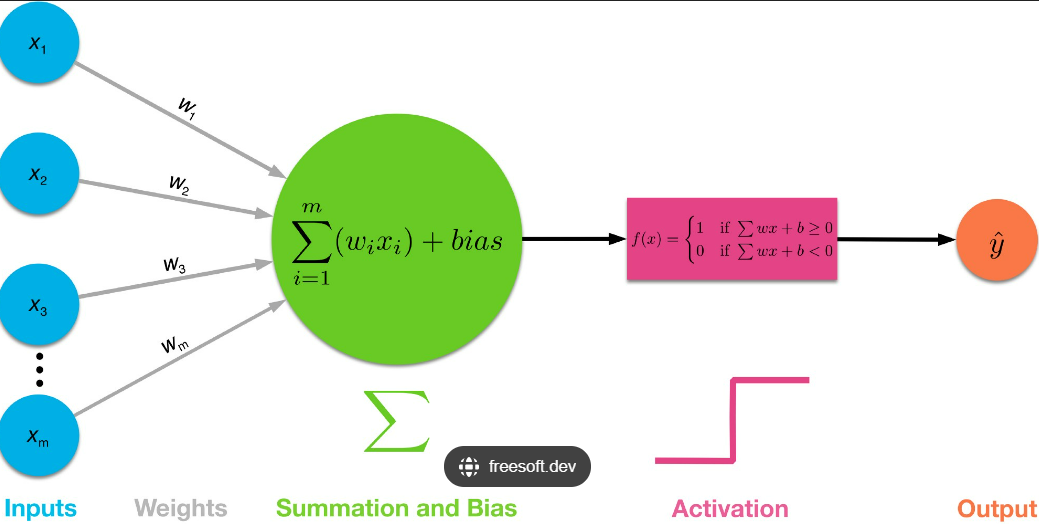
## Fonctionnement du réseau des neurones :

Les réseaux de neurones artificiels souvent appelés « ANN » sont constitués de différentes couches de nœuds (ou neurones artificiels), contenant une couche en entrée, une ou plusieurs couches cachées et une couche en sortie. Chaque nœud, se connecte à un autre et possède un poids et un seuil associés. Si la sortie d'un nœud est supérieure à la valeur de seuil spécifiée, ce nœud est activé et envoie des données à la couche suivante du réseau. Sinon, aucune donnée n'est transmise à la couche suivante du réseau. Chaque nœud individuel peut être considéré comme étant son propre modèle de régression linéaire, composé de données d'entrée, de poids, d'un biais (ou d'un seuil) et d'une sortie.



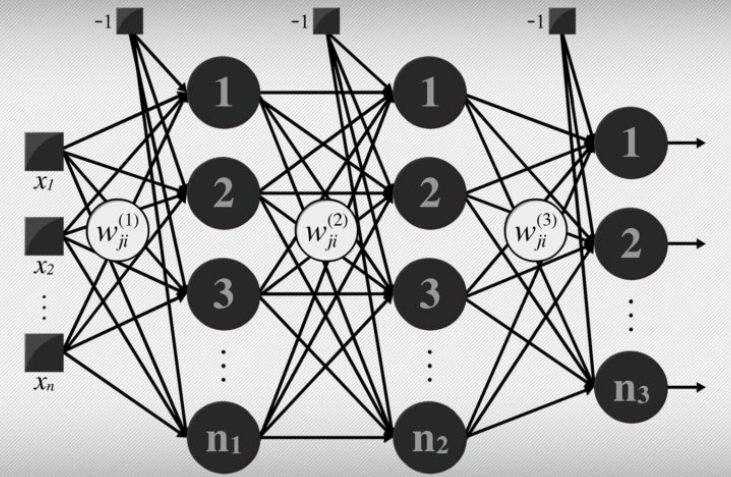
* + 1. **Le perceptron :**

Le perceptron peut être vu comme le type de [réseau de neurones](https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones) le plus simple. C'est un [classifieur linéaire](https://fr.wikipedia.org/wiki/Classifieur_lin%C3%A9aire). Ce type de réseau neuronal ne contient aucun cycle (il s'agit d'un [réseau de neurones à propagation avant](https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_%C3%A0_propagation_avant)). Dans sa version simplifiée, le perceptron est mono-couche et n'a qu'une seule sortie (booléenne) à laquelle toutes les entrées (booléennes) sont connectées. Plus généralement, les entrées peuvent être des nombres réels.Un perceptron à n entrées () et à une seule sortie o est défini par la donnés de de n poids (ou coefficients synaptiques) ()et un biais (ou seuil) θ.

****

* + 1. **Le Perceptron multicouches MLP :**

Un perceptron multicouche (Multilayer Perceptron - MLP) est un réseau neuronal artificiel à réaction qui génère un ensemble de sorties à partir d'un ensemble d'entrées. Un MLP est caractérisé par plusieurs couches de nœuds d'entrée connectés sous forme de graphe orienté entre les couches d'entrée et de sortie. Le MLP utilise la rétro-propagation pour former le réseau. MLP est une méthode d'apprentissage en profondeur. L’intuition derrière les réseaux de neurones est la suivante : si je ne peux pas résoudre un problème avec un seul neurone ou perceptron, je pourrais sans doute le faire en connectant plusieurs neurones entre eux. Le fonctionnement des réseaux de neurones reste toutefois assez similaire à celui du perceptron.

****

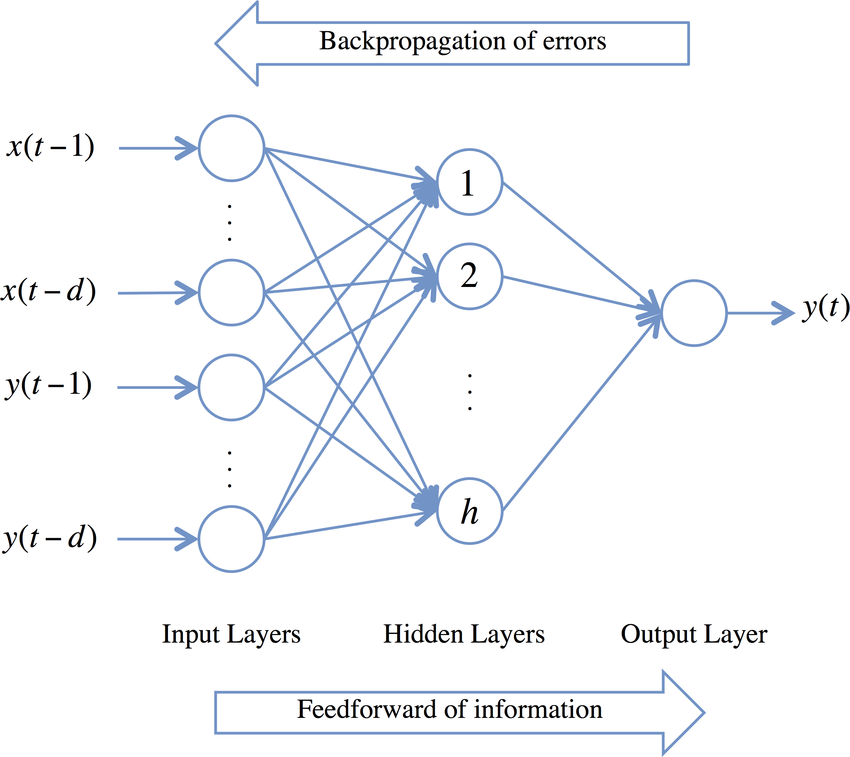
* + - 1. **Propagation directe (Forward propagation):**

Cette technique fait propager les informations à partir de la couche d’entrée, passant par les couches cachées vers la couche de sortie. Une mesure de prédiction est ainsi calculée pour déterminer l’écart entre la valeur prédite et la valeur réelle tout en utilisant la méthode de l’erreur quadratique moyenne ou encore MSE pour obtenir la fonction coût J(a,b) notée par :

Ayant calculé la fonction coût J(a,b), nous passons par la suite à la recherche du minimum globale de cette fonction. Pour ce faire, nous allons utiliser l’algorithme de descente de gradient et plus généralement l’algorithme de descente de gradient stochastique. Une fois que ce calcul est fait, nous passons à la rétropropagation.

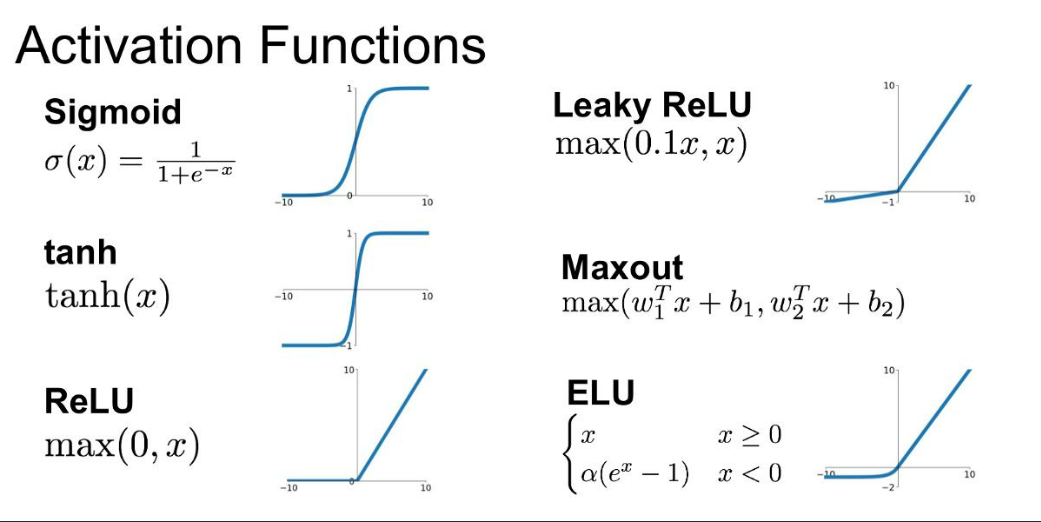
* + - 1. **Rétropropagation (Backpropagation) :**

le flux est inversé de sorte que nous commençons par propager l’erreur à la couche de sortie jusqu’à atteindre la couche d’entrée en passant par la ou les couches cachées et c’est ainsi que les poids sont recalculés et ajustés comme l’illustre l’image se trouvant ci-desssous.



* + - 1. **Fonction d’activation :**

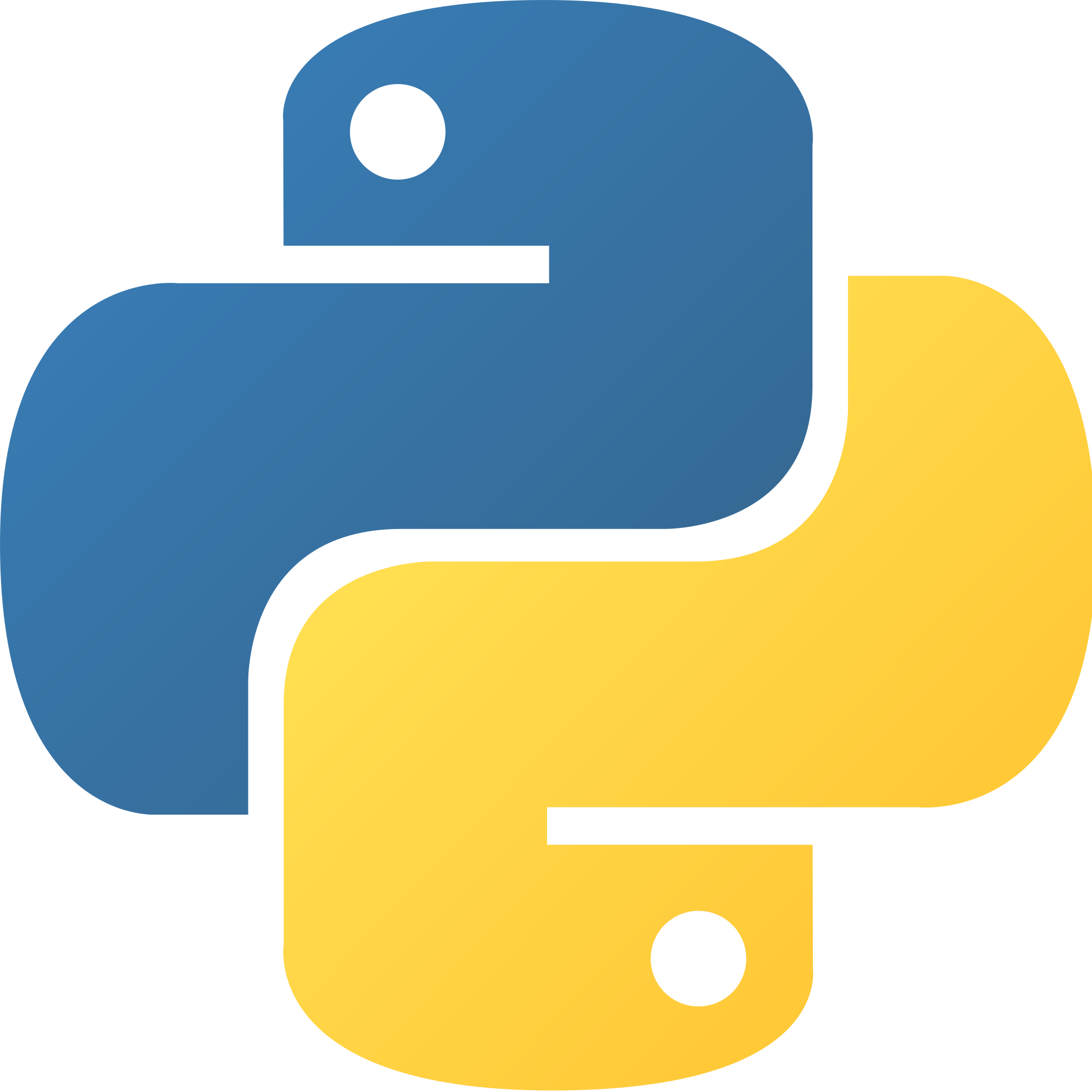
Une fonction d’activation est une fonction mathématique utilisé sur un signal. Elle va reproduire le potentiel d’activation que l’on retrouve dans le domaine de la biologie du cerveau humain. Elle va permettre le passage d’information ou non de l’information si le seuil de stimulation est atteint. Concrètement, elle va avoir pour rôle de décider si on active ou non une réponse du neurone. En effet, il existe plusieurs fonctions d’activation qui sont utilisées dans un réseau ANN. On en distingue :



# Présentation des technologies utilisées dans ce projet :

* 1. **Python :**

Python est un langage de programmation interprété orienté objet. Il est utilisé dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des [bibliothèques](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) spécialisées. En effet, il est largement utilisé dans les applications Web, le développement de logiciels, la science des données et le Machine Learning (ML).



* 1. **La bibliothèque Sklearn :**

C’est une [bibliothèque](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) [Python](https://fr.wikipedia.org/wiki/Python_(langage)) destinée à l'[apprentissage automatique](https://fr.wikipedia.org/wiki/Apprentissage_automatique). Elle comprend notamment des fonctions pour estimer des [forêts aléatoires](https://fr.wikipedia.org/wiki/For%C3%AAts_al%C3%A9atoires), des [régressions logistiques](https://fr.wikipedia.org/wiki/R%C3%A9gression_logistique), des [algorithmes de classification](https://fr.wikipedia.org/wiki/Classification_automatique), et les [machines à vecteurs de support](https://fr.wikipedia.org/wiki/Machine_%C3%A0_vecteurs_de_support). Elle est conçue pour s'harmoniser avec d'autres [bibliothèques](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) [libres](https://fr.wikipedia.org/wiki/Logiciel_libre) [Python](https://fr.wikipedia.org/wiki/Python_(langage)), notamment [NumPy](https://fr.wikipedia.org/wiki/NumPy) et [SciPy](https://fr.wikipedia.org/wiki/SciPy).

****

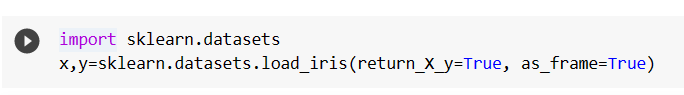
# Présentation des jeux de données utilisés :

# 

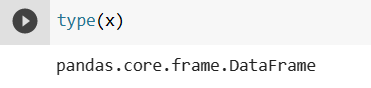
Pour ce projet, nous allons utiliser le célèbre jeu de données IRIS. Ce dataset décrit les espèces d’IRIS par quatre propriétés : longueur et largeur de sépales ainsi que longueur et largeur de pétales. IRIS comporte 150 observations (50 observations par espèce).

* Importation du dataset :

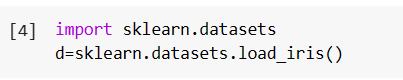
(Méthode 1 : sous forme d’un DataFrame)



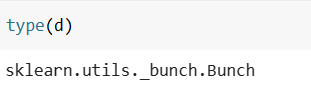
* Type : DataFrame :



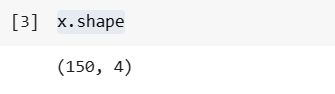
(Méthode 2 : sous forme d’un Bunch)



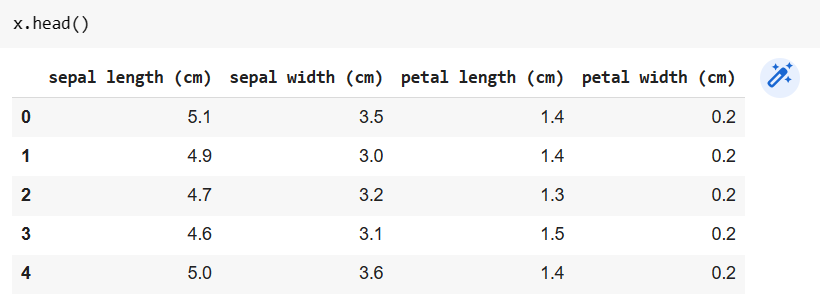
* Type : Bunch



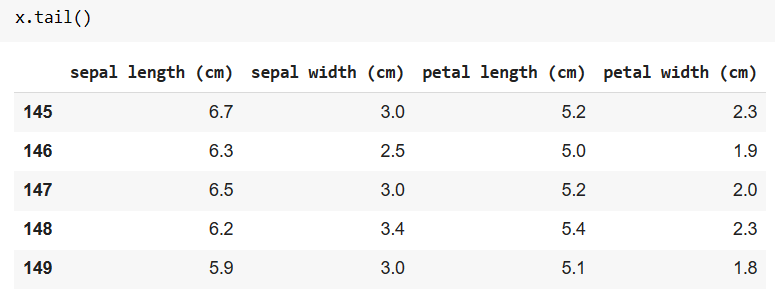
* Affichage des dimensions du dataset :



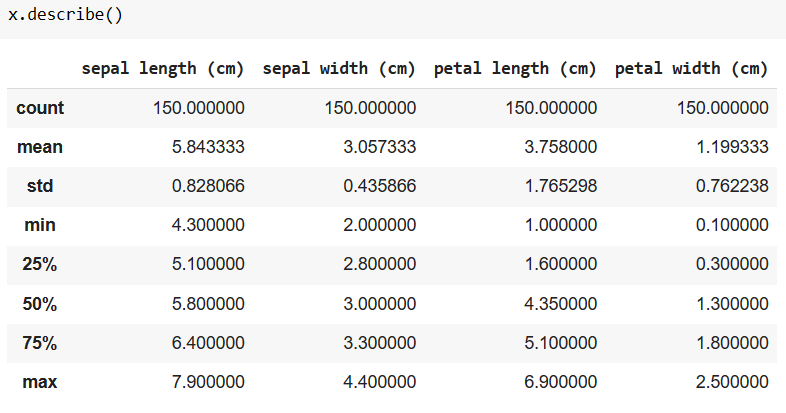
* Ce dataset contient 150 observations et 4 colonnes
* Affichage des 5 premières lignes du dataset :



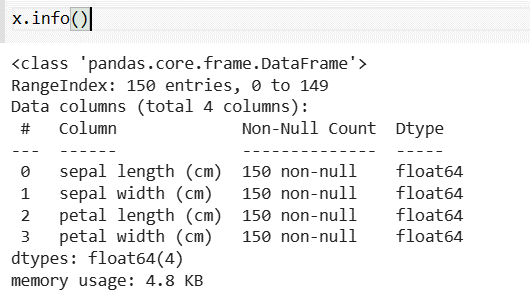
* Affichage des 5 dernières lignes du Dataset :



* Affichage des mesures statistiques du Dataset ( variance, écart type, les quntiles, ...) :

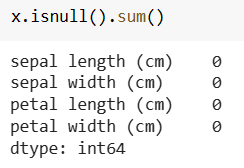


* Affichage des mesures statistiques du Dataset avec la méthode info :

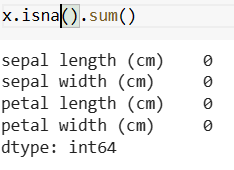


* Affichage de la somme des valeurs null dans chaque colonne :

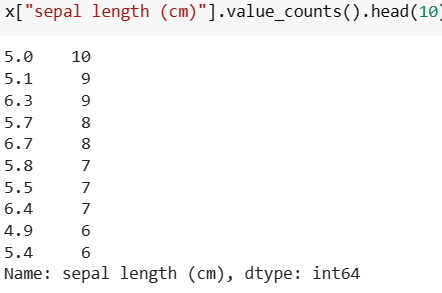
(méthode 1 : avec la méthode isnull() )

 🡺 Iris ne contient AUCUNE valeur nulle

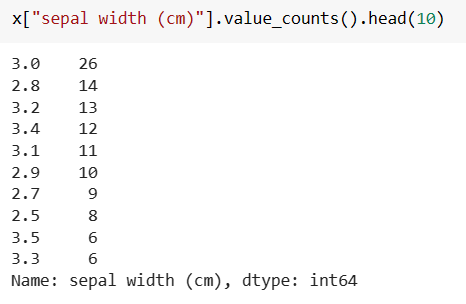
(Méthode 2 : avec la méthode isna() ) :



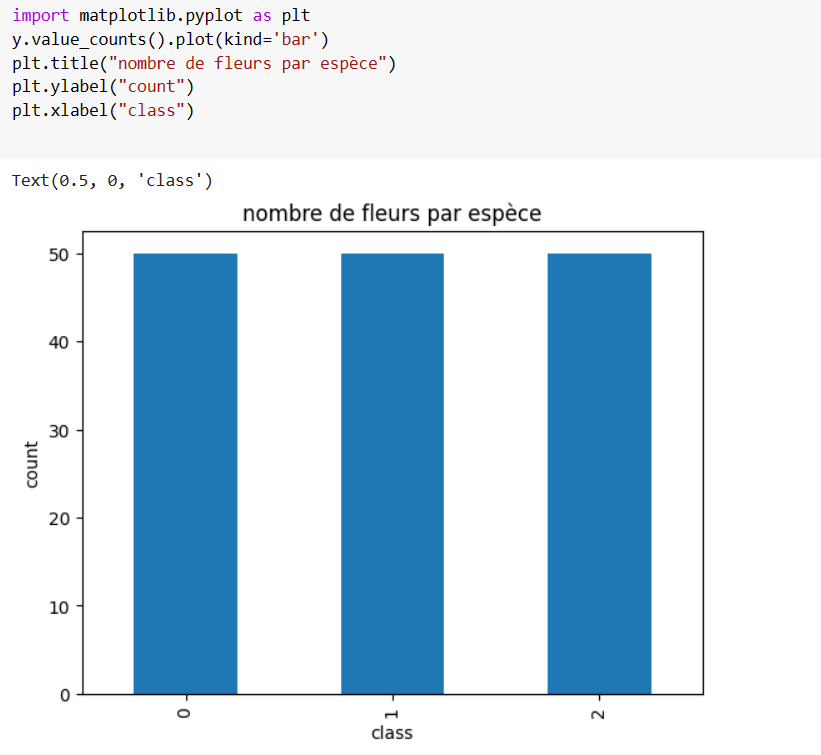
* Calcul du nombre de valeurs des longueurs de sépales :



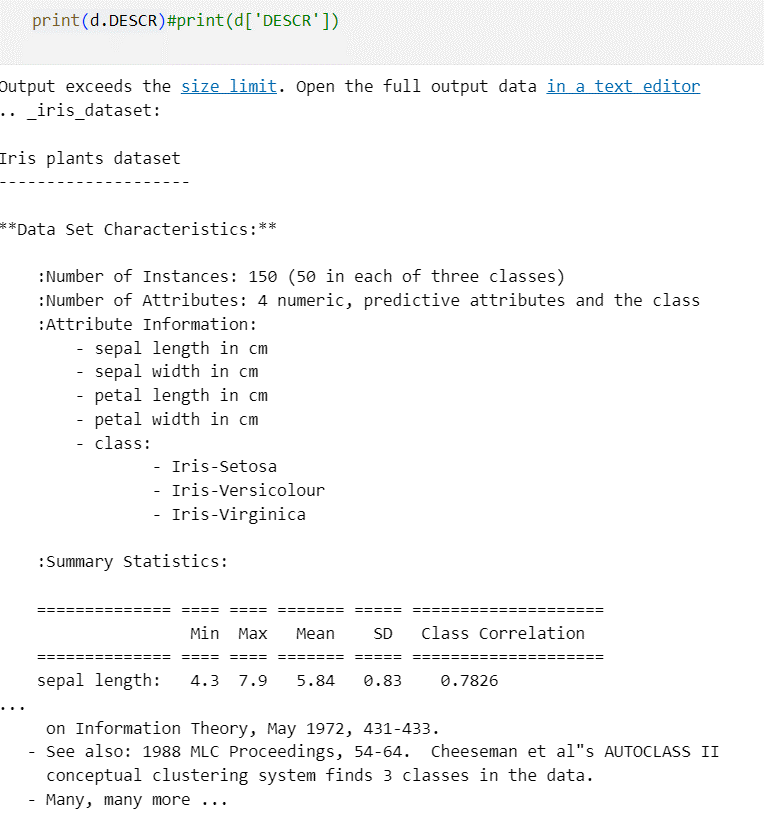
* Calcul du nombre de valeurs des largeurs des sépales :



* Affichage de nombre de fleurs par espèce/ classe :



* Autres informations : ( A partir d’un Bunch)



# Présentation du jeu de données digits :

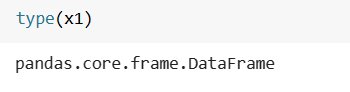
# 

Ce jeu de données est constitué des images de chiffres de 0 jusqu’à 9 comme le montre la figure se trouvant si dessus) contenant 1797 observations et 64 colonnes. Chaque colonne désigne une couleur d’un pixel bien déterminé.

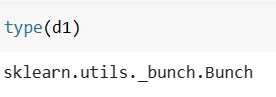
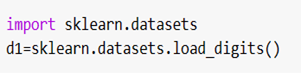
* Chargement du dataset :

Méthode 1 : sous forme d’un DataFrame

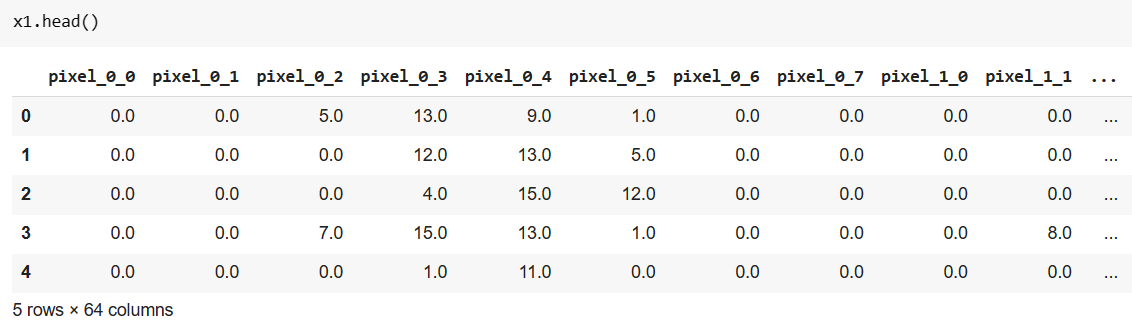




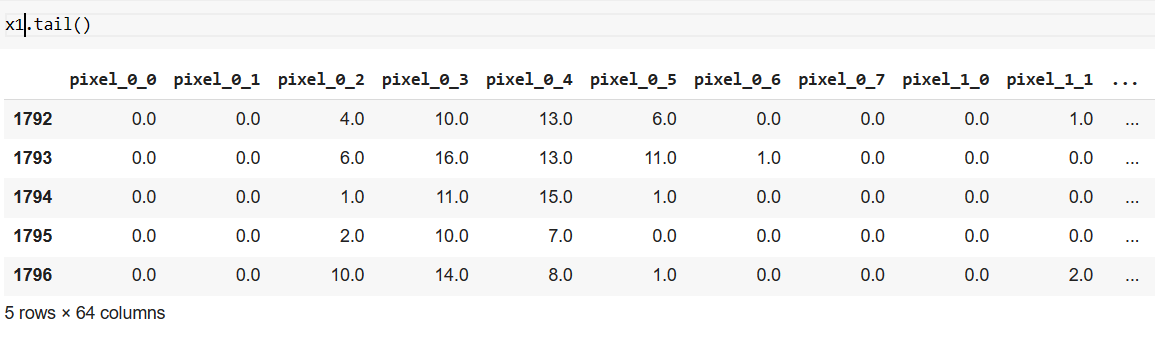
Méthode 2 : sous forme d’un Bunch



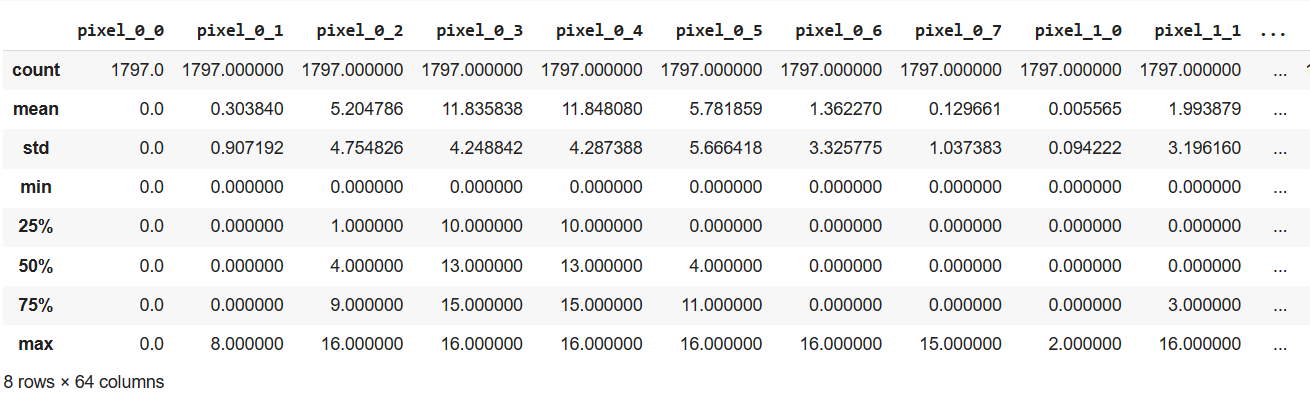
* Visualisation des 5 premières lignes du dataset :



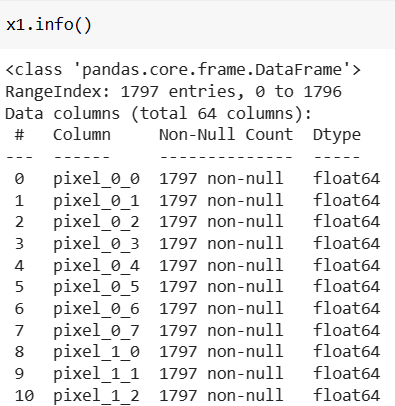
* Visualisation des 5 dernières lignes du dataset :



* Affichage des mesures statistiques du Dataset ( variance, écart type, les quntiles, ...) :

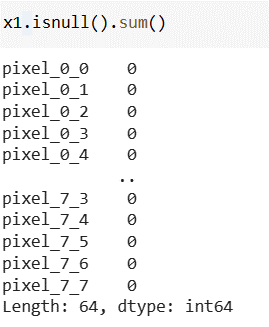


* Affichage des mesures statistiques du Dataset avec la méthode info :

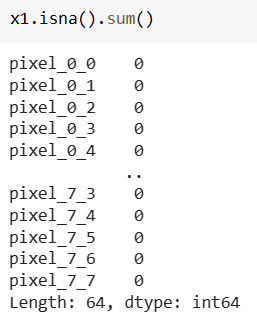


* Affichage de la somme des valeurs nulles dans chaque colonne :

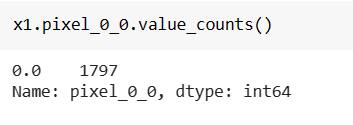
Méthode 1: avec la méthode isnull :



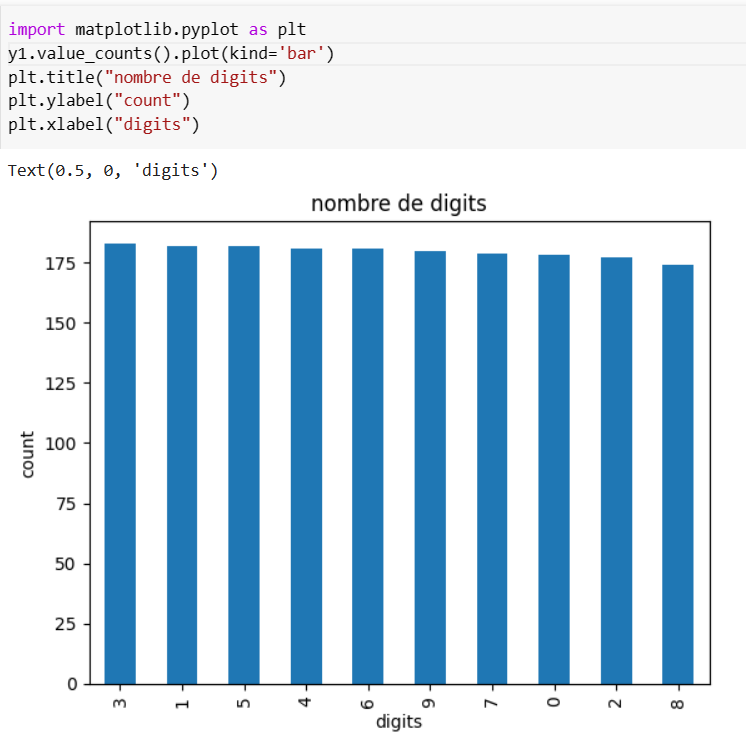
Méthode 2 : avec la fonction isna:



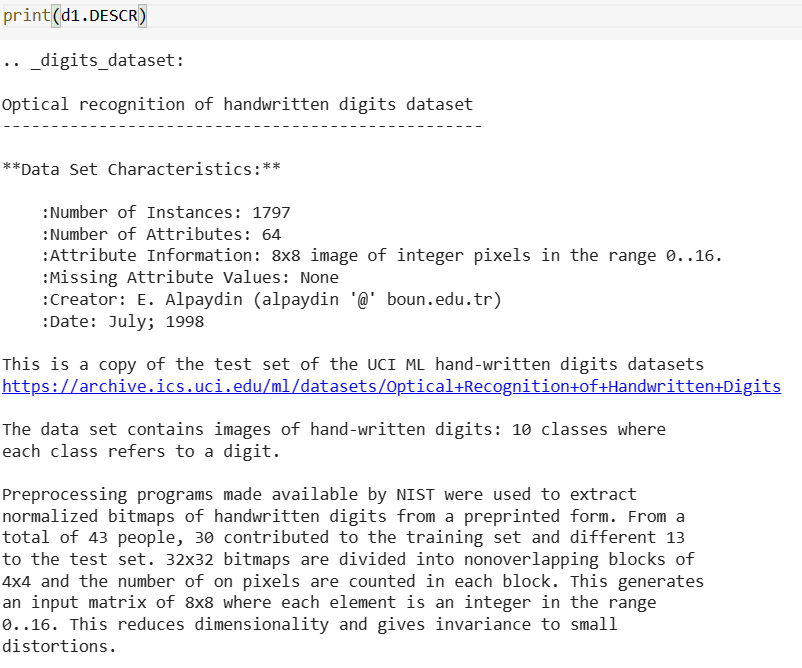
* Calcul du nombre de valeurs de pixel\_0\_0 :



* Affichage du nombre des digits :



* Autres informations :



**Chapitre III :**

**Présentation des résultats et évaluation du modèle :**

**Introduction :**

Dans ce chapitre, nous allons présenter dans un premier temps les résultats obtenus après avoir fait de l’apprentissage avec MLPclassifier repsectivement sur le jeux de données iris et digits de Sklearn et puis nous les analyserons dans un second afin d’évaluer le modèle.

# Présentation des résultats obtenus en utilisant le jeu de données digits :

# Phase d’apprentissage :

# Au cours de cette phase nous allons importer les bibliothèques nécessaires, puis nous allons charger le jeu de données digit. Par la suite, une séparation des données en données d’apprentissage et des données de test sera faite. Finalement nous allons instancier le modèle MLPClassifier afin de pouvoir l’utiliser.

# 

# Prédiction :

# Dans cette phase nous allons prédire y\_pred à partir des données de x\_test.

# 

# Test et évaluation du modèle :

# Score de l’apprentissage et de test :

# Dans cette cellule, nous avons calculé le score d’apprentissage et le score de test.

# En effet, ce paramètre représente le degré de précision moyenne dans l’apprentissage et pendant le test.

# 

# Affichage des poids et des biais :

# Affichage des poids utilisés entre la 1ère et la 2ème couche :

# 

# (B) Affichage du poids w0 :

# 

# Affichage des biais :

# 

# Affichage du meilleur score obtenu pendant l'apprentissage:

# 

# Calcul de la précision, de la justesse, du rappel ainsi que la matrice de confusion:

# Définition de la précision :

# La précision est utilisée dans les problèmes de classification pour indiquer le pourcentage de prédictions positives correctes faites par un modèle. Elle mesure le nombre de prédictions positives correctes faites par un modèle par rapport au nombre total de prédictions positives faites. Elle est calculée par la formule suivante :

# 

# Définition de la justesse : La justesse (ou encore Accuracy en terminologie anglophone) mesure l’efficacité d’un modèle à prédire correctement à la fois les individus positifs et négatifs. Elle est définit par :

# 

# Définition du rappel:

Le rappel permet de savoir le pourcentage de positifs bien prédit par notre modèle. En d’autres termes c’est le nombre de positifs bien prédit (Vrai Positif) divisé par l’ensemble des positifs (Vrai Positif + Faux Négatif). Sa formule est donnée par :

# Définition d’une matrice de confusion :

# La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle. Les prédictions justes et fausses sont révélées et réparties par classe, ce qui permet de les comparer avec des valeurs définies.

# Résultats de calcul :

# 

# 

# 

# Affichage de la courbe loss\_curve:

# 

# Affichage de la courbe learning\_curve :

# 

# Courbe de validation :

# 

# Rapport de classification :

# Un rapport de classification est une mesure d’évaluation des performances dans l’apprentissage automatique. Il est utilisé pour montrer la précision, le rappel, le score F1 et le support de votre [modèle](https://thecleverprogrammer.com/2020/11/27/machine-learning-algorithms-with-python/) de classification entraîné.

# 

# Mesure de l’erreur de classification (MSE) :

# L’erreur quadratique moyenne (MSE) permet de mesurer le degré d’erreur dans les modèles statistiques. Grâce à cette technique on peut évaluer la différence quadratique moyenne entre les valeurs observées et [prévues](https://statisticsbyjim.com/glossary/fitted-values/). Lorsqu’un modèle ne comporte aucune erreur, le MSE est égal à zéro. À mesure que l’erreur de modèle augmente, sa valeur augmente. L’erreur quadratique moyenne est également connue sous le nom d’écart quadratique moyen (MSD) et elle est exprimée par :

# 

# 

# Optimisation de MLPClassifier par Grid search :

# Apprentissage :

# 

# Test et évaluation :

# Affichage des meilleurs parmaètres :

# 

# Affichage de la moyenne, l’ecart-type, l’alpha, le nombre de couches et le learning\_rate dans chaque itération :

# 

# Rapport de classification :

# 

# La justesse du paramètre learning\_rate\_init pendant chaque itération :

# 

# Présentation des résultats obtenus en utilisant le jeu de données Iris :

# Apprentissage :

# 

# Prédiction :

# 

# Test et évaluation du modèle :

# Affichage du score, de la précision, de la justesse et de la matrice de confusion :

# 

# 

# 

# 

# 

# Affichage des poids et des biais utilisés dans l’apprentissage :

# Affichage des poids :

# 

# Affichage des biais :

# 

# Traçage des courbes :

# Loss\_curve :

# 

# Learning\_curve :

# 

# Validation\_curve :

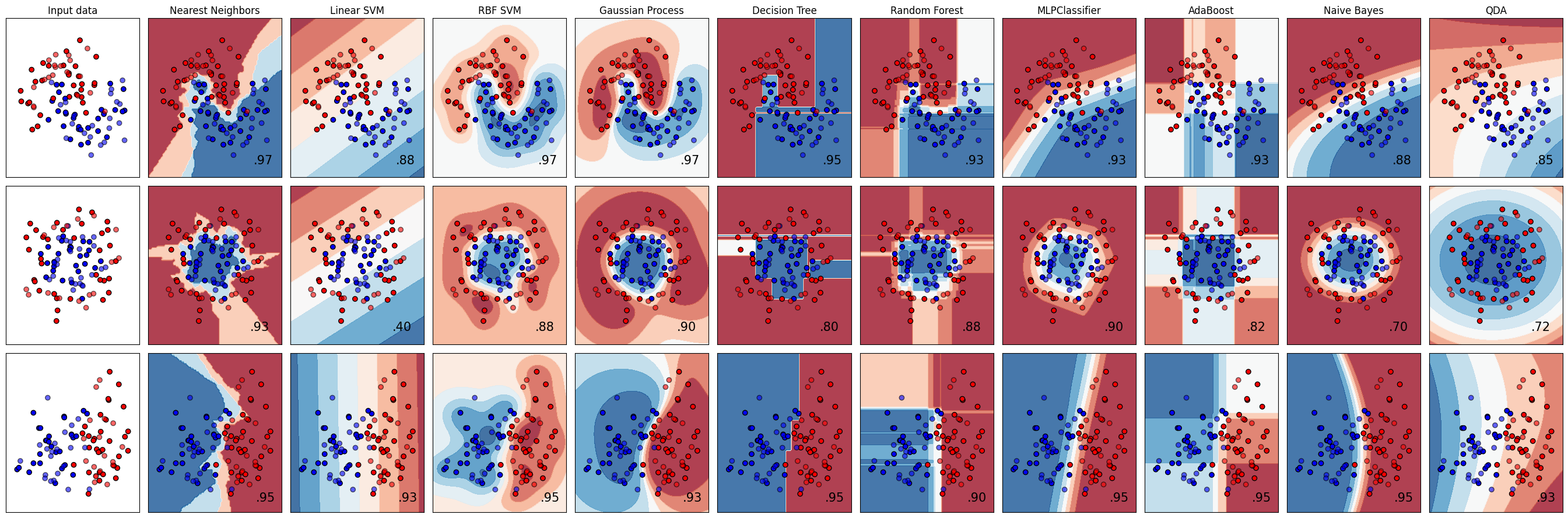
# 

# Etude comparative entre les différents classifieurs sur le même jeu de données :

# Dans cette section, nous allons appliquer plusieurs classifieurs sur le même jeu de données afin de pouvoir comparer le comportement de chaque classifieur. Pour ce faire nous avons utilisé les modèles suivants :

* Nearest Neighbors
* Linear SVM
* RBF SVM
* Gaussian Process
* Decision Tree
* Random Forest
* MLPClassifier
* AdaBoost
* Naive Bayes
* QDA

Afin de pouvoir réaliser cette comparaison nous avons créé une liste dans laquelle nous stockée les noms de modèles. Ainsi, un parcours de la liste a été fait afin de créer des instances de classifieurs. Par la suite, nous fait recours à la méthode fit tout en utilisant le modèle et le jeu de données choisi pour faire de l’apprentissage. L’image se trouvant dans la page suivante résume les résultats de l’apprentissage qu’on avait trouvé :



# Etude comparative entre Iris et digits:

Les objectifs de cette section sont:

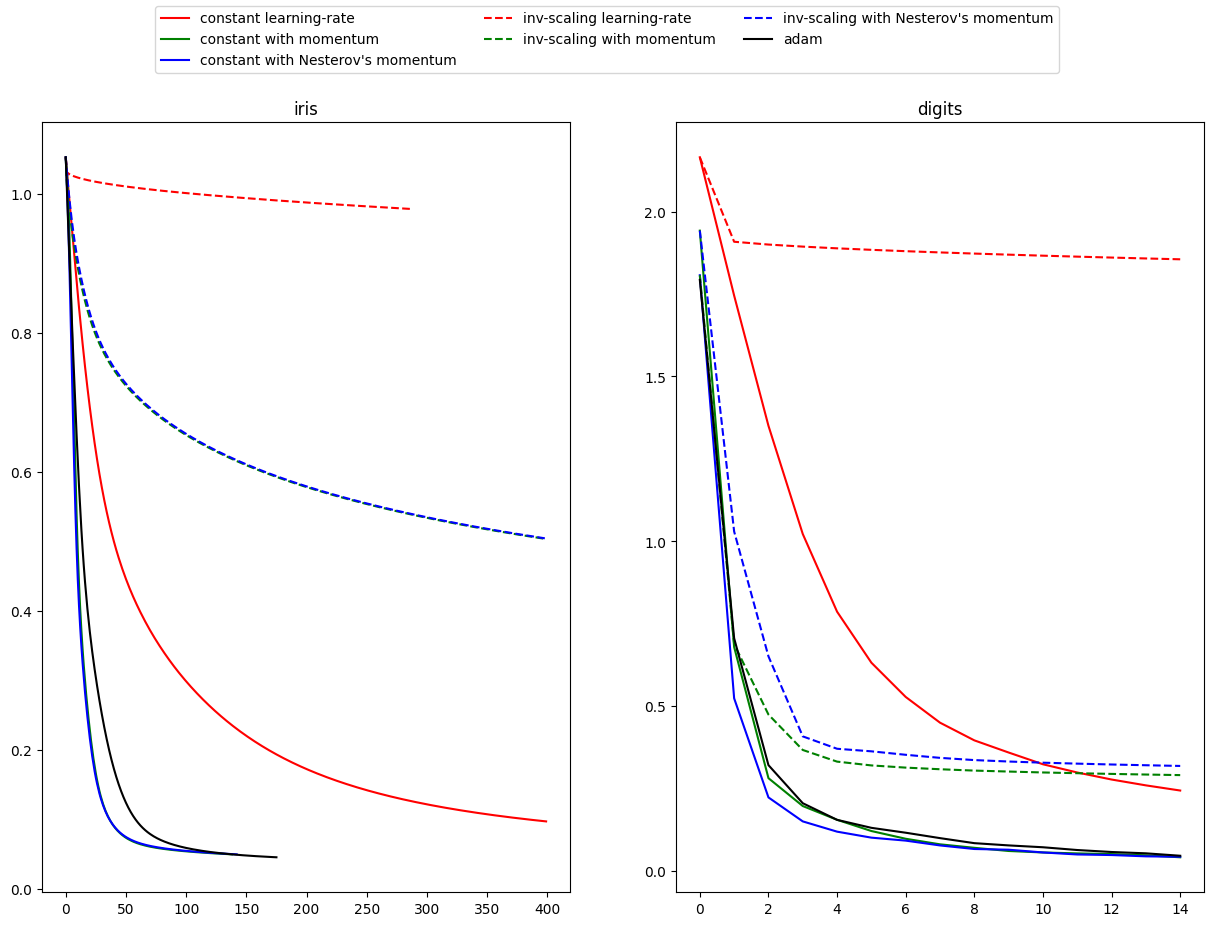
* Etudier le comportement du même modèle utilisé sur le même dataset tout en changeant à chaque fois les hyperparamètres
* Etudier le comportement du même modèle tout en l'appliquant sur différents dataset

1. **Objectif 1:**

Pour ce faire nous avons créé une première liste contenant l'ensemble des hypermètres à utiliser sur le même dataset. De même nous avons créé une deuxième liste "Label" dans laquelle nous avons stocké les "labels" de chaque modèle qu’on lui a changé les hyperparamètres. Ainsi, un parcours de la liste des hyperparamètres est fait pour faire l’instanciation du modèle tout en lui fournissant comme hyperparamètres i -ème élément de la première liste. Par la suite, nous avons tracé les courbes des résultats d’apprentissage.

1. **Objectif 2:**

Pour ce faire nous avons créé une liste contenant les jeux que nous utiliserons dans la phase d’apprentissage. Ces jeux de données sont en fait iris et digits. Après avoir créé cette liste nous avons appliqué le même modèle avec les mêmes hyperparamètres respectivement sur les deux datasets : iris et digits



# Conclusion :

# Dans ce projet, nous avons créé un classifieur perceptron multicouches à partir de MLPClassifier de la bibliothèque sklearn. Ce classifieur a été utilisé sur deux jeux de données que nous avons explorées et présentées. Ce classifieur est composée de plusieurs couches de réseau de neurones. En effet, il contient 3 types de couches : soient une couche d’entrée, un ensemble de couches cachées et finalement une couche de sortie. De même, nous avons mesuré la précision, la justesse et le rappel de ce modèle grâce des métriques bien appropriées. De plus, nous avons clôturé notre projet par une simple comparaison entre l’utilisation de ce même classifieur ( avec les mêmes hyperparamètres) sur deux jeux de données distincts afin de voir son impact sur iris et digits.

.

## 

**Bibliographies :**

[sklearn.neural\_network.MLPClassifier — scikit-learn 1.2.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)

[1.17. Neural network models (supervised) — scikit-learn 1.2.2 documentation](https://scikit-learn.org/stable/modules/neural_networks_supervised.html)

[Multilayer perceptron - Wikipedia](https://en.wikipedia.org/wiki/Multilayer_perceptron)

[Multilayer Perceptron Definition | DeepAI](https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron)

[Multi-Layer Perceptrons Explained and Illustrated | by Dr. Roi Yehoshua | Apr, 2023 | Towards Data Science](https://towardsdatascience.com/multi-layer-perceptrons-8d76972afa2b)

[Le perceptron multicouches | Le perceptron multi-couches - Deep learning - - Kongakura](https://kongakura.fr/article/Le-perceptron-multicouches)

[Intelligence artificielle : définition et utilisation | Actualité | Parlement](https://www.europarl.europa.eu/news/fr/headlines/society/20200827STO85804/intelligence-artificielle-definition-et-utilisation) [européen (europa.eu)](https://www.europarl.europa.eu/news/fr/headlines/society/20200827STO85804/intelligence-artificielle-definition-et-utilisation)

*mlv.fr/~dr/XPOSE2014/Machin\_Learning/D\_Machine\_Learning.html*

[*Machine Learning : Définition, fonctionnement, utilisations*](https://datascientest.com/machine-learning-tout-savoir)[*(datascientest.com)*](https://datascientest.com/machine-learning-tout-savoir)

[*Qu'est-ce que l'intelligence artificielle (IA) ? - France | IBM*](https://www.ibm.com/fr-fr/cloud/learn/what-is-artificial-intelligence)[*Qu’est-ce que l’intelligence artificielle (IA) ? | Oracle Canada*](https://www.oracle.com/ca-fr/artificial-intelligence/what-is-ai/)[*Intelligence artificielle — Wikipédia (wikipedia.org)*](https://fr.wikipedia.org/wiki/Intelligence_artificielle)

[*Qu'est-ce que l'intelligence artificielle (IA) ? - France | IBM*](https://www.ibm.com/fr-fr/cloud/learn/what-is-artificial-intelligence)