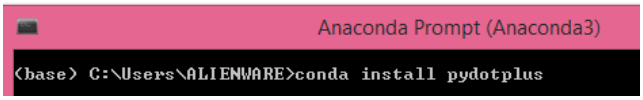
Université **S**idi **M**ohamed **B**en **A**bdellah **E**cole **N**ationale des **S**ciences **A**ppliquées de **F**ès **A.U :** 2023-2024	**Module** : Intelligence Artificielle **Niveau** : 2ème année GSEII **Professeur** : Hiba Chougrad

TP1

- Commencez par installer la version 32 ou 64 bits d'Anaconda
- Après l'installation ouvrez Anaconda Prompt et installez pydotplus



- Une fois terminé, ouvrez Anaconda Prompt et tappez : ***jupyter notebook***

- Accédez au dossier TP1_IA :
- Ouvrez le **1er Lab (*python101.ipynb)*** *qui donne les bases de python*
- Exécutez le premier exemple et expliquez ce que fait le programme.
- Python travaille conjointement avec plusieurs librairies telles que ***Numpy***(cette bibliothèque permet d'effectuer des calculs numériques avec Python. Elle facilite la gestion des tableaux de nombres en utilisant *numpy.array()*). L'exemple 2 explique comment on peut importer la librairie et travailler avec les fonctions qu'elle offre.
  Testez l'exemple avec d'autres fonctions. Ci-dessous un tableau qui liste quelques fonctions :

Commande	Bibl.	Résultat
`sqrt(x)`	numpy	Racine carrée de x
`exp(x), log(x)`	numpy	Exponentielle, logarithme népérien
`sin(x), cos(x)`	numpy	Fonctions sinus et cosinus
`arcsin(x), arccos(x)`	numpy	Fonctions trigonométriques inverses
`rand()`	numpy.random	Générateur aléatoire sur $[0;1]$
`rand(n,p)`	numpy.random	Générateur aléatoire de matrice de dimension $n \times p$ sur $[0;1]$

- Exécutez les exemples 3 et 4 et 5 pour comprendre comment fonctionnent les objets : Listes, Tuples et les Dictionnaires.
- Exécutez l'exemple 6 qui explique comment on définit les fonctions.
- Exécutez les exemples 7 et 8 pour savoir manipuler les structures de contrôles (conditions, boucles)
- Faites l'activité 1

- Ouvrez le **2ème Lab *(PandasTutorial.ipynb)*** qui introduit la librairie **Pandas**. Cette librairie permettra de traiter les tableaux de données (DataFrames) facilement avec Python.
- Exécutez les exemples 1 et 2 et déduisez ce que fait la fonction pd.head( ).
- Exécutez l'exemple 3 et déduisez ce que fait la fonction pd.tail( ).
- Exécutez les exemples 4, 5 et 6 pour savoir afficher la taille d'un tableau de données.
- Exécutez les exemples à partir de 7 jusqu'à 12 pour savoir gérer les lignes et les colonnes d'un tableau de données.
- Exécutez l'exemple 13 pour apprendre à réorganiser les données d'un tableau selon l'ordre de la colonne choisit.
- Exécutez l'exemple 14 pour savoir compter les valeurs uniques de chaque colonne.
- Exécutez l'exemple 15 pour apprendre à visualiser les valeurs d'un tableau de données (colonnes vs lignes)
- Faites l'activité 2

- Ouvrez le **3ème Lab *(MeanMedianMode.ipynb)*** où vous utiliserez des fonctions prédéfinit sur la

librairie Numpy pour calculer la moyenne, la médiane et le mode d'un ensemble de données.

- Exécutez l'exemple 1 et déduisez ce que fait la fonction numpy.random.normal(), puis dire si le résultat obtenu de la moyenne est compatible avec vos attentes.
- Exécutez l'exemple 2 pour apprendre à visualiser les données en histogramme.
- Exécutez l'exemple 3 pour apprendre à calculer la médiane d'un ensemble de données.
- Exécutez les exemples 4 et 5 et essayez d'interpréter les résultats(comparez le Mean et la Médiane).
- Exécuter l'exemple 6 et 7 pour apprendre à calculer le mode d'un ensemble de données.

- Ouvrez le **4ème Lab (*MeanMedianExercice.ipynb*) e**t appliquez les notions apprises au 3ème Lab pour répondre à l'exercice.

- Ouvrez le **5ème Lab (*StdDevVariance.ipynb*)** où vous utiliserez des fonctions prédéfinit dans la librairie Numpy pour calculer l'écart-type(standard deviation) et la variance d'un ensemble de données.
- Exécutez l'exemple 1 pour générer un ensemble de données aléatoirement à partir d'une distribution gaussienne.
- Exécutez les exemples 2 et 3 pour savoir calculer l'écart-type(standard deviation) et la variance resp. d'un ensemble de données.
- *Faites l'activité 3*

- Ouvrez **le 6ème Lab (*Outliers.ipynb*)** où vous apprendrez à alléger l'impact des valeurs aberrantes (Outliers) sur le comportement d'un modèle d'apprentissage :
- Exécutez le 1er exemple pour générer des données aléatoires traduisant les revenus d'une population suivant une loi gaussienne de moyenne 27000 et d'écart-type 15000. Ajoutez à la population une personne ayant 1000000000 comme revenu. Que remarquez-vous après visualisation des données.
- Exécutez le 2eme exemple pour recalculer la nouvelle moyenne des revenus. Quelle remarque pourriez-vous tirer de ce résultat.
- Exécutez le 3eme exemple pour définir la fonction **reject_outliers()** qui ne gardera que les valeurs de revenus issue de l'intervalle **[u-2s, u+2s]** où **u** représente la médiane des revenus et **s** son écart type. Que remarquez-vous après visualisation des données.
- Exécutez le 4eme exemple pour calculer la moyenne des revenus après avoir appliqué le filtre **[u-2s, u+2s].** Quelle remarque pouvez-vous tirer du résultat.
- Faites l'activité.

- Ouvrez le **7ème Lab (*MissingData.ipynb*)** où vous apprendrez à travailler avec des Datasets ayant des valeurs manquantes :
- Exécutez le 1er exemple pour importer *Pima Indians Diabetes Dataset.* Cette Dataset permet de prédire l'apparition du diabète chez les Indiens Pima compte tenu des détails médicaux suivant : Number of times pregnant, Plasma glucose concentration at 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg) ,Triceps skinfold thickness (mm) , 2-Hour serum insulin (mu U/ml) , Body mass index (weight in kg/(height in m)^2) , Diabetes pedigree function and Age (years).
- Appliquez la fonction **describe()** sur la dataset pour afficher ses statistiques descriptifs. Essayez d'extraire les caractéristiques ayant une valeur minimale anormale égale à zéro (si la valeur zéro est associée anormalement à une caractéristique ceci traduit l'absence de la valeur associée à cette caractéristique).
- Exécutez le 2eme exemple pour afficher les 20 premières données de la Dataset en utilisant la fonction **head ()**

- Exécutez le 3eme exemple pour calculer le nombre de valeurs manquantes traduites par zéro en chaque colonne n'admettant pas zéro comme valeur logique. Interprétez les résultats.
- Exécutez le 4eme exemple pour remplacer les valeurs manquantes par NaN en utilisant la fonction **fillna().**
- Exécutez le 5eme exemple pour enlever toutes les lignes ayant les valeurs manquantes en utilisant la fonction **dropna()** prédéfinit dans Pandas. Comparez le nombre de données sans et avec des valeurs manquantes. Dire si cette approche est bénéfique pour résoudre le problème des valeurs manquantes. Justifiez votre réponse.
- Exécutez le 6eme exemple pour remplacer les valeurs manquantes d'une caractéristique (colonne) par la moyenne des valeurs de la même caractéristique.
- Faites l'activité.

- Ouvrez le **8ème Lab (*RescalingData.ipynb*)** pour apprendre à normaliser et standardiser les données :
- Exécutez le 1er exemple pour importer la Dataset Iris. Iris est une dataset qui comprend 50 échantillons de trois espèces d'iris (*Iris setosa*, *Iris virginica* et *Iris versicolor*). Quatre features ont été mesurées pour chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres.
- Utilisez la fonction **normalize()** du package **preprocessing** de la bibliothèque **scikit-learn** pour normaliser les valeurs de la base de données entre 0 et 1 . Affichez les résultats.
- Dans le 2eme exemple utilisez la fonction **scale()** prédéfinit sur **scikit-learn.**
- Faites l'activité.

- Ouvrez le **9ème Lab (*MatPlotLib.ipynb*)** où vous apprendrez les outils de bases de la librairie ***MatPlotLib***. La librairie *MatPlotLib* permet de générer des graphiques dans des formats différents sous Python.
- Exécutez les exemples 1 et 2 pour apprendre à tracer une ou plusieurs courbes à la fois dans une seule interface.
- Exécutez l'exemple 3 pour apprendre à sauvegarder les images générées par la librairie *MatPlotLib*.
- Exécutez les exemples restant pour découvrir les différents graphiques que vous pouvez générer avec la librairie *MatPlotLib*.
- Faites l'activité 4

- Ouvrez le **10ème Lab (*Seaborn.ipynb*)** où vous allez découvrir la librairie de visualisation **Seaborn** qui est une version amélioré de *MatPlotLib*
- Dans l'exemple 1 exécutez le code pour importer la dataset indiquée, puis visualisez en utilisant *MatPlotLib* le de nombre de voitures modèle 2019 en fonction des vitesses(Gears) de leur boites à vitesse.
- Exécutez l'exemple 2 pour importer Seaborn
- Dans l'exemple 3 vous allez exécuter le même code que l'exemple 1 en utilisant Seaborn à la place de *MatPlotLib*. Comparer la qualité des deux figures données par Seaborn et MatPlotLib.
- Continuez à exécuter les cellules du Lab et interprétez à chaque fois les figures obtenues.