Atelier 1 : Installation scikit-learn, chargement et visualisation des données

I. Installation :

1. Installation Python 3 (Langage de programmation pour Data Science)

```
$ sudo apt update
$ sudo apt -y upgrade
$ sudo apt install python3
```

Vérifier l'installation

```
$ python3 --version
```

2. Installation PIP3 (Installateur de packages pour Python3)

```
$ sudo apt install python3-pip
```

Vérifier l'installation

```
$ pip3 --version
```

Utilisation de PIP3 pour l'installation des packages

`$ pip install packages-name` (Exemple : `pip install numpy`)

3. Installation PyCharm (Editeur python)

```
$ sudo snap install pycharm-community --classic
```

4. Installation scikit-learn pour LINUX/Ubuntu (Framework de Data Science)

Disponible sur : https://scikit-learn.org/stable/index.html

```
$ pip3 install -U scikit-learn
```

Vérifier l'installation

```
$ python3 -m pip show scikit-learn
```

5. Installation OPENCV (Bibliothèque de vision par ordinateur)

```
$sudo apt update
$sudo apt install libopencv-dev python3-opencv
```

Pour l'installation depuis la source, vous pouvez consulter le lien :

https://linuxize.com/post/how-to-install-opencv-on-ubuntu-20-04/

II. Chargement et visualisation des données :

1. Chargement des données

- *Exemple de chargement des données manuellement :*

```
import numpy as np
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
Y = np.array([1, 1, 1, 2, 2, 2])
```

- *Chargement des datasets (jeu de données) publics :*
  Voir le lien https://scikit-learn.org/stable/datasets.html, pour quelques exemples.
  Remarque : Il n'y a pas de moyen standard pour charger les dataset dans scikit-learn, chaque dataset a sa propre méthode de chargement selon le type et la structuration des données qu'il contient (Nous traiterons cette partie dans les prochains ateliers).

2. Visualisation des données

La visualisation des données permet d'avoir une idée sur la distribution de ces données dans l'espace, et aussi sur l'espace des hypothèses.

Exemple de code python :

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import neighbors, datasets

# import some data to play with
iris = datasets.load_iris()
# we only take the first two features
# slicing by using a two-dim dataset
X = iris.data[:, :2]
y = iris.target
# Create color maps
cmap_bold = ['darkorange', 'red', 'darkblue']
# Put the result into a color plot
plt.figure(figsize=(8, 6))
# Plot the training points
sns.scatterplot(x=X[:, 0], y=X[:, 1], hue=iris.target_names[y],
          palette=cmap_bold, alpha=1.0, edgecolor="black")
plt.title("Data visualisation")
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.show()
```