Introduction à Python

Apprentissage des bases de la programmation avec Python

Qu'est-ce que Python?

- Langage de programmation interprété, de haut niveau
- Créé par Guido van Rossum et publié en 1991
- Utilisé pour le développement web, l'analyse de données, l'intelligence artificielle, et plus

Installer Python

- Téléchargement depuis python.org
- Installation de l'IDE (ex: PyCharm, VS Code, Jupyter)

Syntaxe de base en Python

- Les variables et les types de données : int, float, str, bool
- Exemples de base :

```
x = 5
y = 3.14
nom = 'Alice'
```

Structures de contrôle

- Les boucles : for, while
- Les conditions : if, elif, else

```
if x > 0:
  print('x est positif')
else:
  print('x est négatif ou nul')
```

Les fonctions en Python

- Définition et appel de fonctions
- Arguments et valeurs de retour

```
def ajouter(a, b):
    return a + b

resultat = ajouter(3, 4)
```

Listes et Dictionnaires

- Listes : collections ordonnées et modifiables

```
fruits = ['pomme', 'banane', 'cerise']
```

Dictionnaires : collections non ordonnées de paires clé/valeur

```
capitales = {'France': 'Paris', 'Italie': 'Rome'}
```

Modules et bibliothèques

- Importation de modules : import math, import os
- Utilisation de bibliothèques populaires : numpy, pandas, matplotlib

```
import math
print(math.sqrt(16))
```

Modules et bibliothèques

Qu'est-ce que Scikit-Learn ?

•**Scikit-Learn** est une bibliothèque open source pour le langage de programmation Python.

•Elle est largement utilisée pour **l'apprentissage automatique**.

•Basée sur **NumPy**, **SciPy** et **matplotlib**.

Fonctionnalités Principales

•**Classification**: identifier la catégorie à laquelle un exemple appartient.

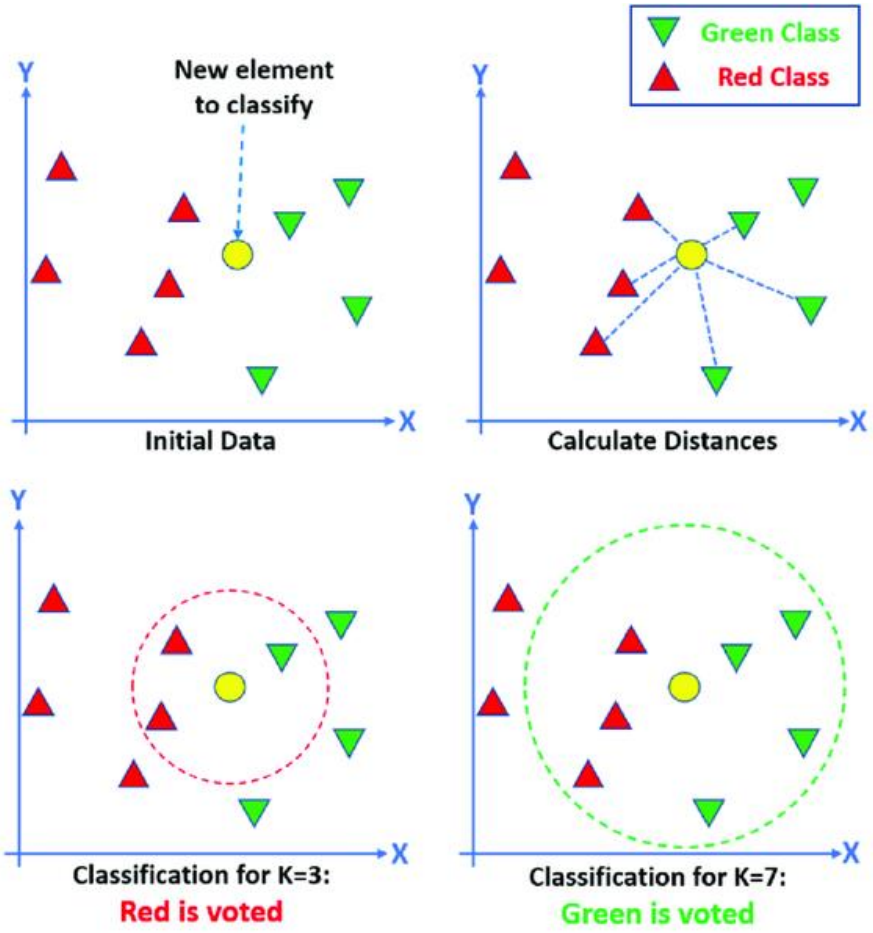
•**Régression**: prédire une valeur continue.

•**Clustering**: grouper les exemples sans étiquettes préexistantes.

•**Réduction de dimension**: réduire le nombre de variables aléatoires.

•**Sélection de modèle**: comparer, valider et choisir les paramètres de modèle.

Modules et bibliothèques (KNN)



Modules et bibliothèques

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target
feature_names = iris.feature_names
target_names = iris.target_names
print("Feature names:", feature_names)
print("Target names:", target_names)
print("\nFirst 10 rows of X:\n", X[:10])
```

Modules et bibliothèques

```
# Installer scikit-learn si ce n'est pas déjà fait
# !pip install scikit-learn

# Importer les bibliothèques nécessaires
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
```

Modules et bibliothèques

```
# Charger les données Iris
iris = load_iris()
X = iris.data  # Caractéristiques (features)
y = iris.target  # Labels (cible)

# Afficher les premières lignes des données pour mieux
comprendre
print("Caractéristiques (features) :\n", X[:5])
print("Labels (cible) :\n", y[:5])

# Diviser les données en ensembles d'entraînement (70%) et
de test (30%)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

Modules et bibliothèques

```
# Initialiser le modèle k-NN avec k=3
knn = KNeighborsClassifier(n_neighbors=3)

# Entraîner le modèle sur les données d'entraînement
knn.fit(X_train, y_train)

# Afficher le modèle entraîné
print("\nModèle entraîné :\n", knn)
```

Modules et bibliothèques

```
# Prédire les labels pour l'ensemble de test
y_pred = knn.predict(X_test)

# Afficher les prédictions
print("\nPrédictions :\n", y_pred)

# Calculer et afficher la précision du modèle
accuracy = accuracy_score(y_test, y_pred)
print(f"\nPrécision du modèle : {accuracy:.2f}")
```

Conclusion et ressources

- Récapitulatif des concepts appris
- Ressources pour aller plus loin :
- - Documentation officielle : https://docs.python.org/3/
- - Cours en ligne : Codecademy, Coursera, Udemy
- - Communautés : Stack Overflow, Reddit