

TP 01: Classification with scikit-learning

Objective

In this exercise, we introduce a **supervised classification model** using **scikit-learn**. Students will learn how to:

- Handle a real dataset
- Train a classification model
- Evaluate its performance
- Interpret results

Problem Description

We want to classify flowers into different species based on their characteristics using the famous **Iris dataset**.

Each flower is described by:

- Sepal length
- Sepal width
- Petal length
- Petal width

The goal is to predict the **species** of the flower.

Step 1: Import Libraries

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

Step 2: Load the Dataset

```
iris = datasets.load_iris()

X = iris.data      # Features
y = iris.target    # Labels
```

Step 3: Split the Dataset

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.3, random_state=42  
)
```

Step 4: Train the Model (KNN)

```
model = KNeighborsClassifier(n_neighbors=3)  
model.fit(X_train, y_train)
```

Step 5: Make Predictions

```
y_pred = model.predict(X_test)
```

Step 6: Evaluate the Model

```
accuracy = accuracy_score(y_test, y_pred)  
print("Accuracy:", accuracy)
```

Step 7: Simple Visualization (Optional)

```
plt.scatter(X[:, 0], X[:, 1], c=y)  
plt.xlabel("Sepal Length")  
plt.ylabel("Sepal Width")  
plt.title("Iris Dataset Visualization")  
plt.show()
```