**Assignment 0. Human Action Recognition using Machine Learning**

**student information**

Student ID: m5281054

Name: WANG Binghao

**Required Libraries:**

import os  
import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd from sklearn.neighbors import KNeighborsClassifier

**Run the Program:**

Python: Python3.12

I have put the executable file into ITA09A\_ssignment\_0\Otherfiles\code

This file also includes all data sets. Make sure you open this folder and install the required libraries. Click Run.

**About Code Explanation:**

The code primarily achieves the following key functionalities:

**Data Visualization:** It uses the Matplotlib library to visualize the training data, with each figure containing data from three axes.

**Feature Extraction:** RMS (Root Mean Square) features are extracted from the training data and utilized for training the classifier.

**Model Training:** The KNeighborsClassifier from the sklearn library is employed to train the k-nearest neighbors classifier.

**Model Prediction:** The trained model is used to predict classes for the test data.

**Result Visualization:** The predicted results are visualized using Matplotlib and Pandas libraries, where a table of predictions is generated and displayed.

the key parts of the code and their purposes:

**1．Data Visualization : This section reads the data from each file, parses it, and plots the figures for each axis (X, Y, Z) for each action and file.**

for action in actions:

action\_folder = os.path.join(train\_data\_folder, action)

for i in range(1, 8):

# Reading and parsing data from the file

data = file.readlines()

data = [list(map(float, line.strip().split('\t'))) for line in data]

data = list(zip(\*data))

# Plotting the data for each axis (X, Y, Z) for each action and each file

plt.figure(figsize=(8, 6))

plt.plot(data[0], color='red', label='X')

plt.plot(data[1], color='green', label='Y')

plt.plot(data[2], color='blue', label='Z')

plt.title(f'{action} - {i:02d}')

plt.legend()

plt.show()

**2. Feature Extraction:** **This part calculates the RMS features for each axis (X, Y, Z) of each data file for training.**

for action in actions:

action\_features = []

for i in range(1, 8):

# Reading and parsing data from the file

data = file.readlines()

data = [list(map(float, line.strip().split('\t'))) for line in data]

data = np.array(data)

# Calculating RMS values for each axis

rms\_values = np.sqrt(np.mean(data \*\* 2, axis=0))

action\_features.append(rms\_values)

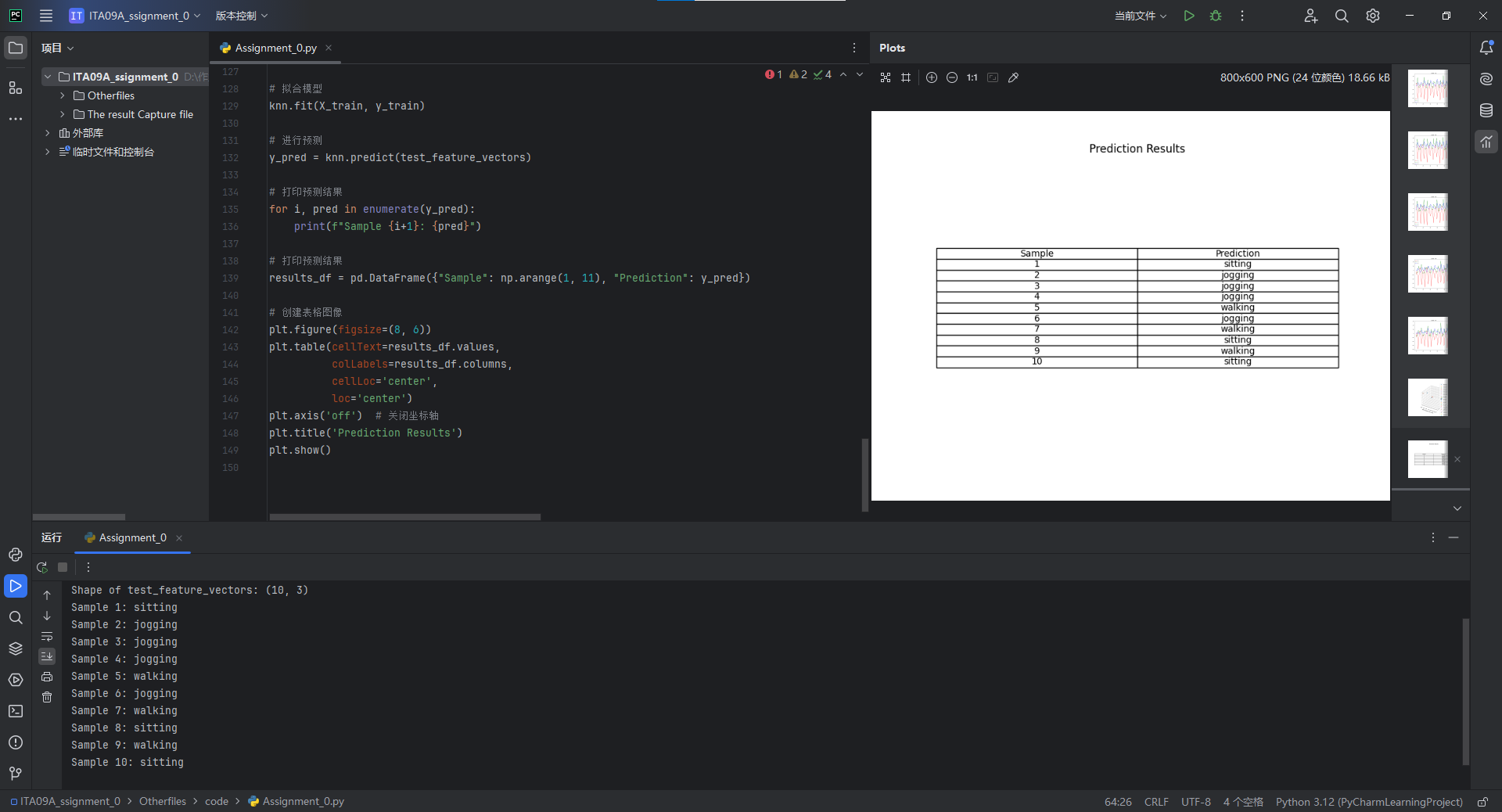
feature\_vectors.append(action\_features)

**3. Model Training: This section creates a k-nearest neighbor classifier with 3 neighbors and trains the model using the extracted features and corresponding labels.**

knn = KNeighborsClassifier(n\_neighbors=3)

knn.fit(X\_train, y\_train)

**Final effect (screenshot from my own experiment)**

****