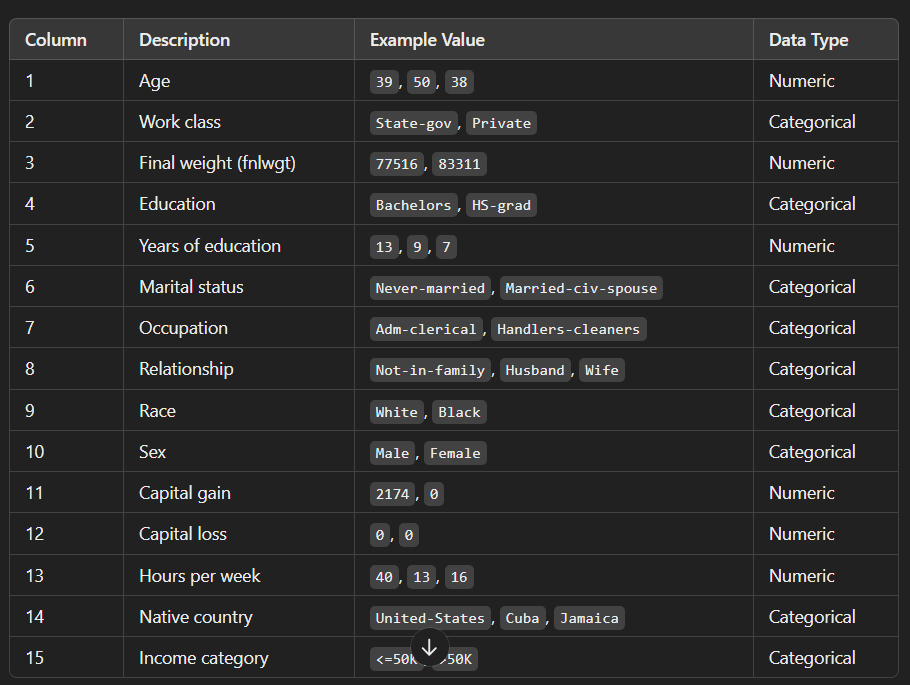
**ЛАБОРАТОРНА РОБОТА № 2**

**Мета роботи**: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити різні методи класифікації даних та навчитися їх порівнювати.

Github: [link](https://github.com/Ajoke32/AILabs)

**Завдання 1.** Класифікація за допомогою машин опорних векторів (SVM)

1. Випишіть у звіт всі 14 ознак з набору даних – їх назви та що вони позначають та вид (числові чи категоріальні).



1. Обчисліть значення інших показників якості класифікації (акуратність, повнота, точність) та разом з F1 занесіть їх у звіт.

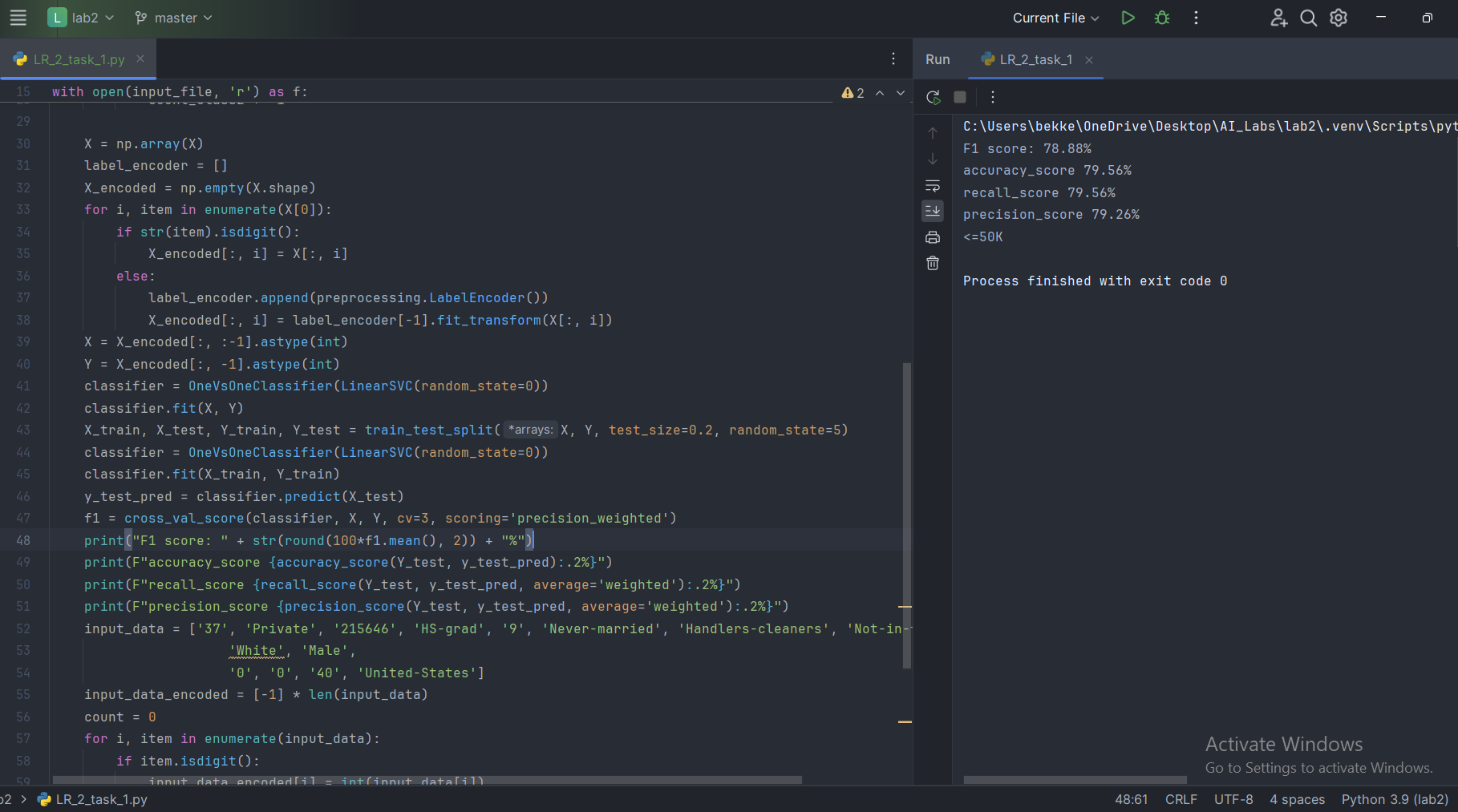


Рис 1. Розрахунки

1. Код програми занесіть у звіт.
2. import numpy as np  
   from sklearn import preprocessing  
   from sklearn.svm import LinearSVC  
   from sklearn.multiclass import OneVsOneClassifier  
   from sklearn.model\_selection import train\_test\_split, cross\_val\_score  
   from sklearn.metrics import accuracy\_score, recall\_score, precision\_score  
     
   input\_file = 'income\_data.txt'  
     
   X = []  
   Y = []  
   count\_class1 = 0  
   count\_class2 = 0  
   max\_datapoints = 25\_000  
   with open(input\_file, 'r') as f:  
    for line in f.readlines():  
    if count\_class1 >= max\_datapoints and count\_class2 >= max\_datapoints:  
    break  
    if '?' in line:  
    continue  
    data = line[:-1].split(', ')  
    if data[-1] == '<=50K' and count\_class1 < max\_datapoints:  
    X.append(data)  
    count\_class1 += 1  
     
    if data[-1] == '>50K' and count\_class2 < max\_datapoints:  
    X.append(data)  
    count\_class2 += 1  
     
    X = np.array(X)  
    label\_encoder = []  
    X\_encoded = np.empty(X.shape)  
    for i, item in enumerate(X[0]):  
    if str(item).isdigit():  
    X\_encoded[:, i] = X[:, i]  
    else:  
    label\_encoder.append(preprocessing.LabelEncoder())  
    X\_encoded[:, i] = label\_encoder[-1].fit\_transform(X[:, i])  
    X = X\_encoded[:, :-1].astype(int)  
    Y = X\_encoded[:, -1].astype(int)  
    classifier = OneVsOneClassifier(LinearSVC(random\_state=0))  
    classifier.fit(X, Y)  
    X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=5)  
    classifier = OneVsOneClassifier(LinearSVC(random\_state=0))  
    classifier.fit(X\_train, Y\_train)  
    y\_test\_pred = classifier.predict(X\_test)  
    f1 = cross\_val\_score(classifier, X, Y, cv=3, scoring='precision\_weighted')  
    print(print("F1 score: " + str(round(100\*f1.mean(), 2)) + "%"))  
    print(F"accuracy\_score {accuracy\_score(Y\_test, y\_test\_pred):.2%}")  
    print(F"recall\_score {recall\_score(Y\_test, y\_test\_pred, average='weighted'):.2%}")  
    print(F"precision\_score {precision\_score(Y\_test, y\_test\_pred, average='weighted'):.2%}")  
    input\_data = ['37', 'Private', '215646', 'HS-grad', '9', 'Never-married', 'Handlers-cleaners', 'Not-in-family',  
    'White', 'Male',  
    '0', '0', '40', 'United-States']  
    input\_data\_encoded = [-1] \* len(input\_data)  
    count = 0  
    for i, item in enumerate(input\_data):  
    if item.isdigit():  
    input\_data\_encoded[i] = int(input\_data[i])  
    else:  
    input\_data\_encoded[i] = int(label\_encoder[count].transform([input\_data[i]])[0])  
    count += 1  
    input\_data\_encoded = np.array(input\_data\_encoded).reshape(1, 14)  
    predicted\_class = classifier.predict(input\_data\_encoded)  
    print(label\_encoder[-1].inverse\_transform(predicted\_class)[0])

Рис 1. Код програми

1. Зробіть висновок до якого класу належить тестова точка

**Тестова точка, відноситься до класу людей, у яких дохід менше або рівне 50К**

**Завдання 2.** Порівняння якості класифікаторів SVM з нелінійними ядрами