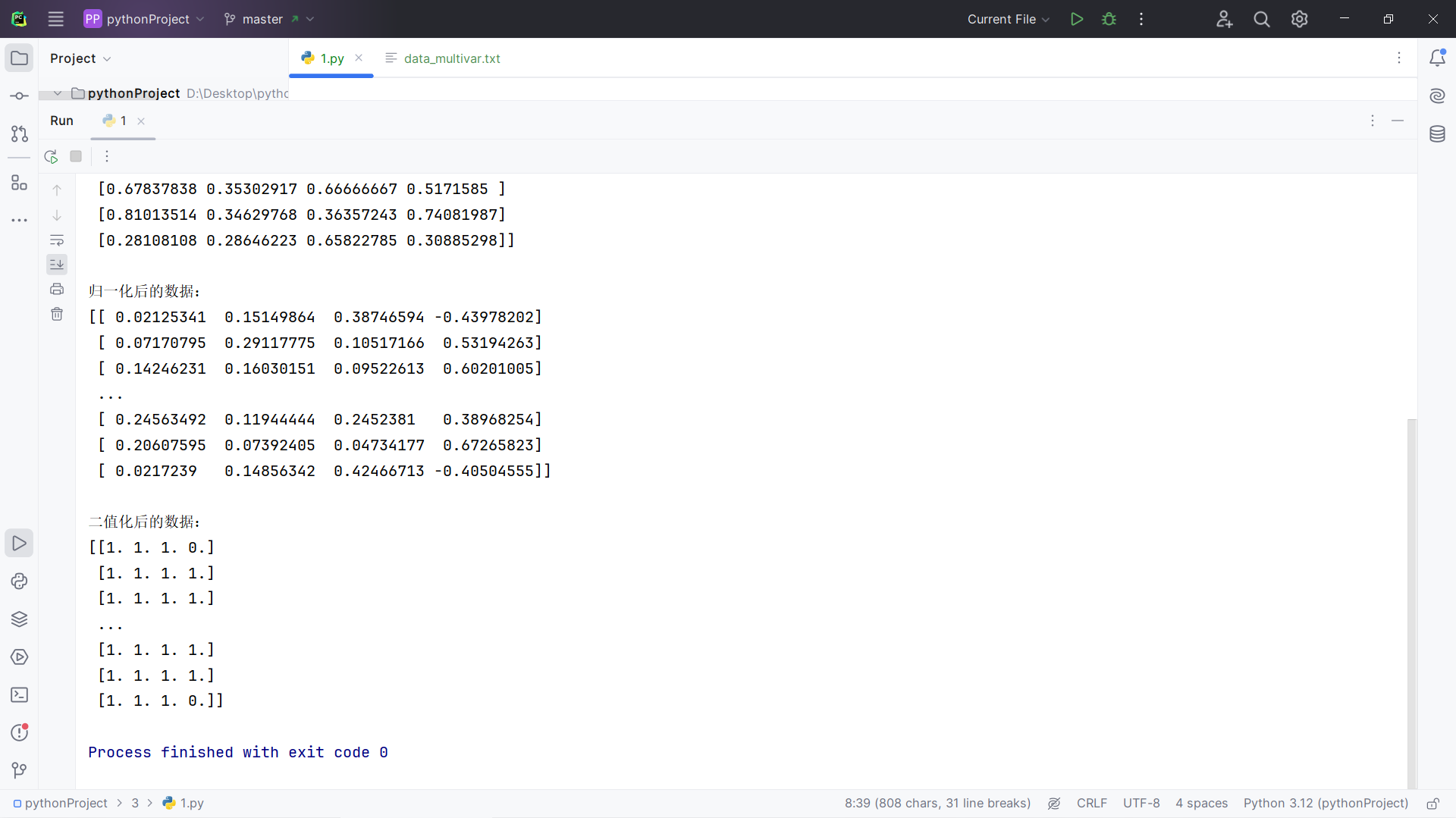
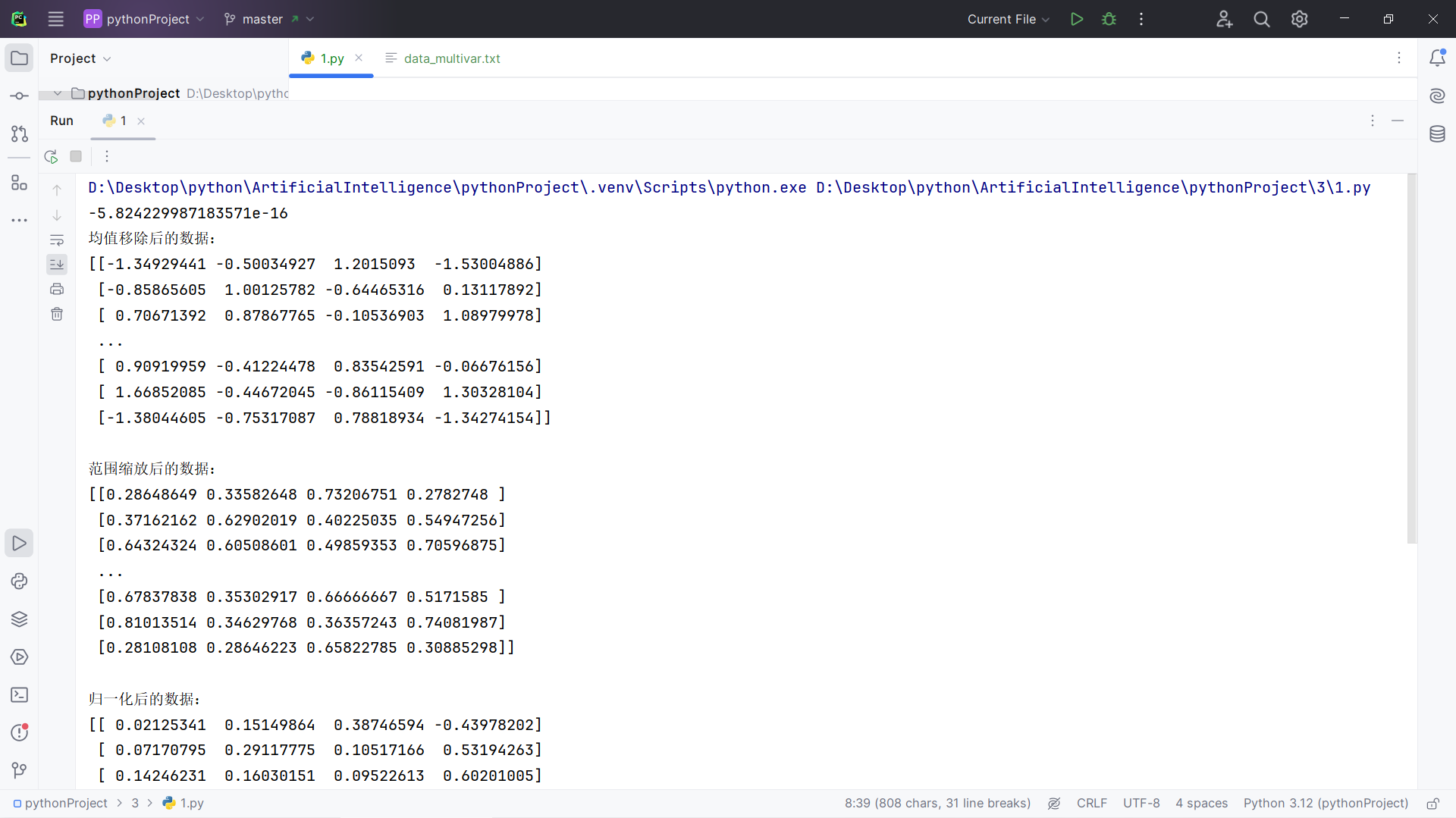
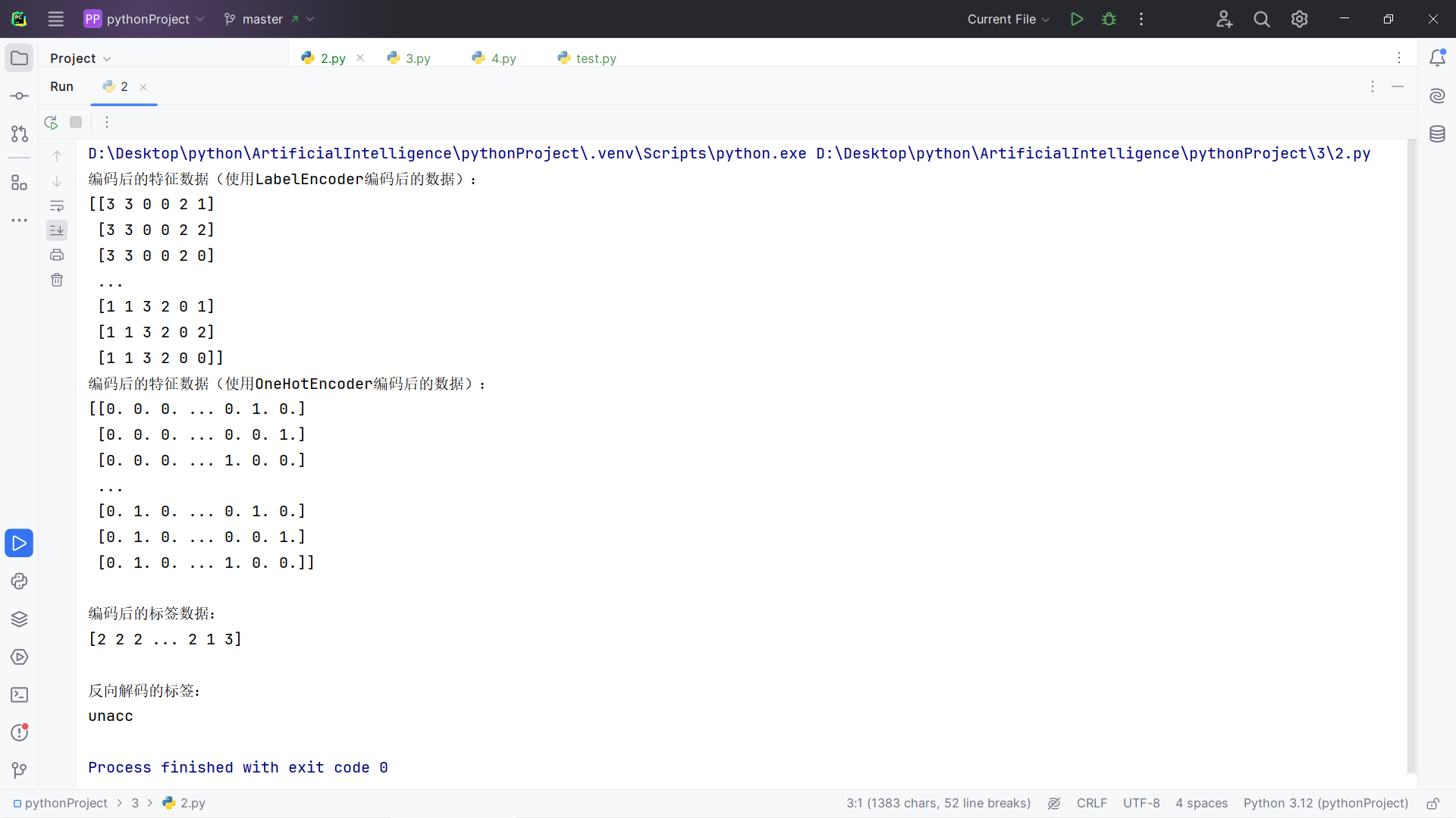
1、读取data\_multivar.txt数据，输出样本特征的均值移除、范围缩放、归一化、二值化（阈值为均值）结果

from sklearn import preprocessing  
import numpy as np  
  
# 读取数据  
data = np.loadtxt('data\_multivar.txt', delimiter=',')  
  
# 均值移除（Mean Removal）  
data\_standard = preprocessing.scale(data)  
# print(data\_standard.mean(axis=0))  
# 范围缩放（Min-Max Scaling）  
data\_scaler = preprocessing.MinMaxScaler(feature\_range=(0, 1))  
data\_scaled = data\_scaler.fit\_transform(data)  
  
# 归一化（Normalization）  
data\_normalized = preprocessing.normalize(data, norm='l1')  
  
# 二值化（Binarization）  
threshold = np.mean(data\_standard.mean(axis=0), axis=0)  
# print(threshold)  
data\_binarizer = preprocessing.Binarizer(threshold=threshold)  
data\_binarized = data\_binarizer.transform(data)  
  
# 打印结果  
print("均值移除后的数据：")  
print(data\_standard)  
print("\n范围缩放后的数据：")  
print(data\_scaled)  
print("\n归一化后的数据：")  
print(data\_normalized)  
print("\n二值化后的数据：")  
print(data\_binarized)



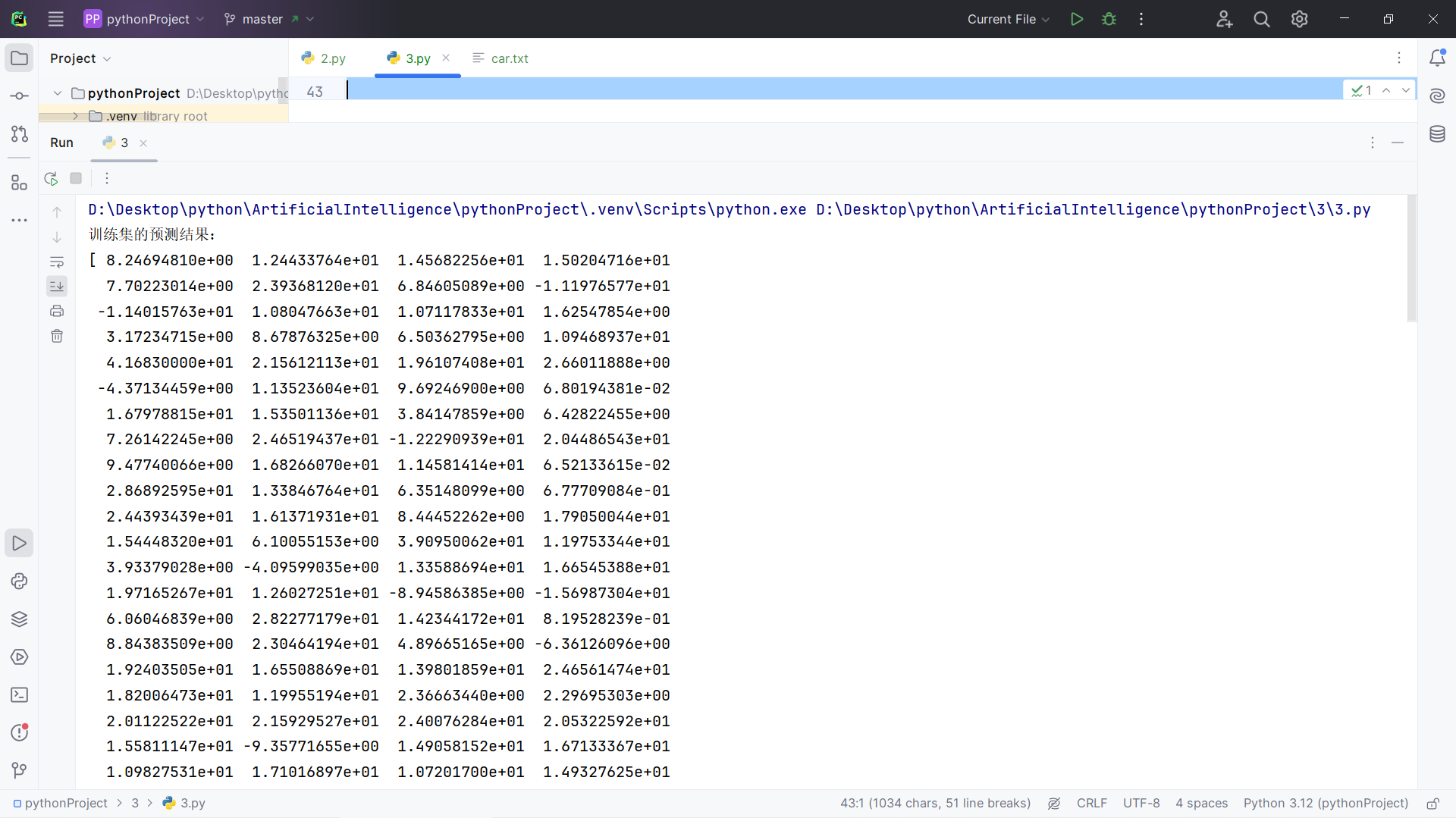
2、读取car.txt数据，将样本的所有特征及最后一列进行labelEncoder和OnehotEncoder编码，输出编码后的数据，并输入任意一个样本反向解码

import numpy as np  
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  
  
# 读取数据  
with open('car.txt', 'r') as file:  
 lines = file.readlines()  
  
# 提取特征和标签  
data = []  
labels = []  
for line in lines:  
 line = line.strip().split(',')  
 data.append(line[:-1]) # 提取特征  
 labels.append(line[-1]) # 提取标签  
  
data = np.array(data)  
labels = np.array(labels)  
  
# 对特征进行 LabelEncoder 编码和 OneHotEncoder 编码  
label\_encoders = []  
data\_encoded1 = None  
for i in range(data.shape[1]):  
 label\_encoder = LabelEncoder()  
 data\_encoded\_column = label\_encoder.fit\_transform(data[:, i])  
 if data\_encoded1 is None:  
 data\_encoded1 = data\_encoded\_column.reshape(-1, 1)  
 else:  
 data\_encoded1 = np.hstack((data\_encoded1, data\_encoded\_column.reshape(-1, 1)))  
 label\_encoders.append(label\_encoder)  
  
onehot\_encoder = OneHotEncoder()  
data\_encoded2 = onehot\_encoder.fit\_transform(data\_encoded1)  
  
# 对标签进行 LabelEncoder 编码  
label\_encoder\_label = LabelEncoder()  
labels\_encoded = label\_encoder\_label.fit\_transform(labels)  
  
# 输出编码后的数据  
print("编码后的特征数据（使用LabelEncoder编码后的数据）：")  
print(data\_encoded1)  
  
print("编码后的特征数据（使用OneHotEncoder编码后的数据）：")  
print(data\_encoded2.toarray())  
  
print("\n编码后的标签数据：")  
print(labels\_encoded)  
  
# 反向解码示例  
sample\_index = 0 # 假设要解码的是第一个样本  
decoded\_label = label\_encoder\_label.inverse\_transform([labels\_encoded[sample\_index]])[0]  
print("\n反向解码的标签：")  
print(decoded\_label)

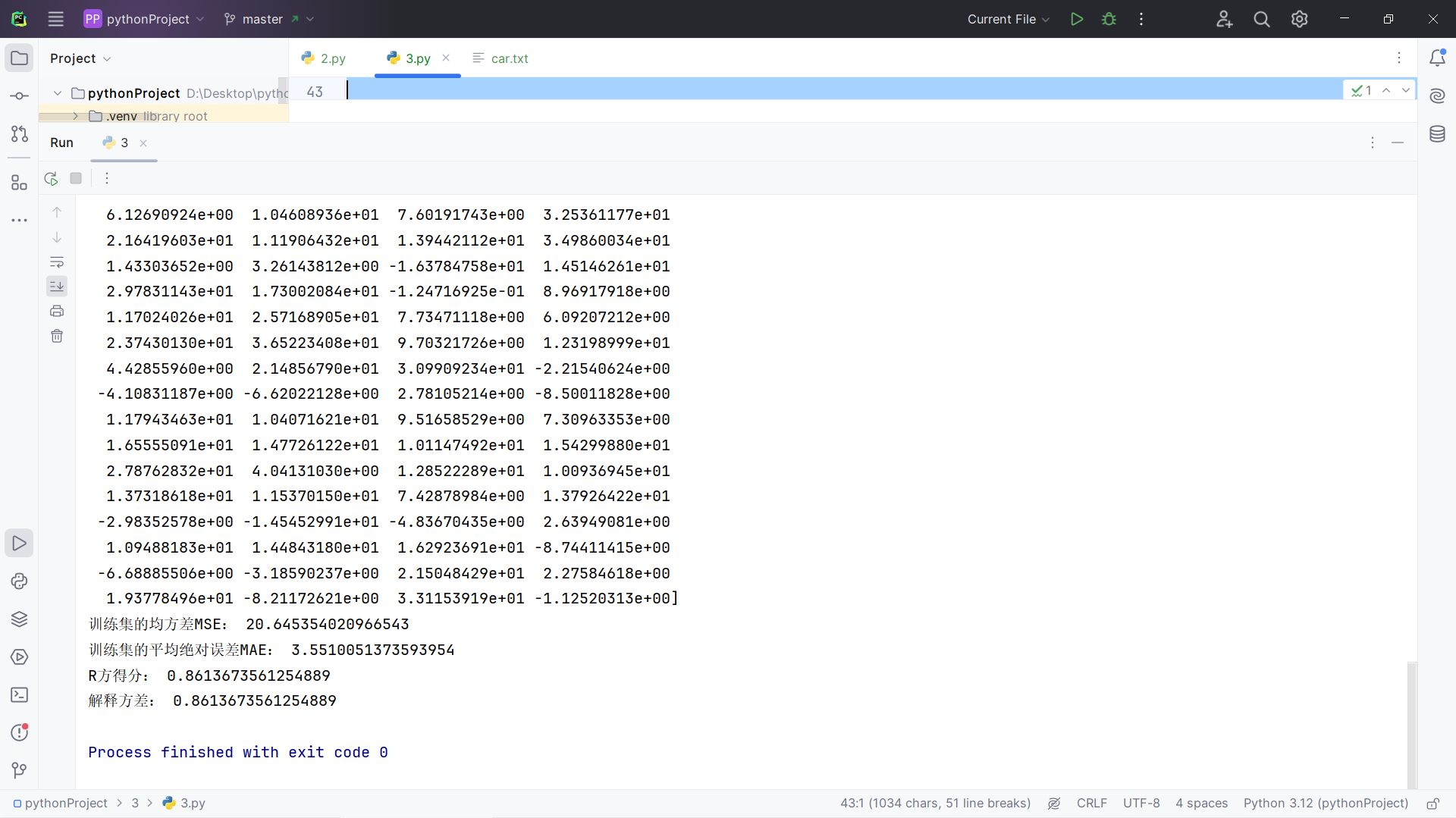


3、读取data\_multivar.txt,先对数据进行训练集和测试集拆分（0.2比例），使用线性回归模型对训练集进行训练，输出训练集的预测结果，并输出训练集的均方差MSE，平均绝对误差MAE，R方得分，解释方差

from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score, explained\_variance\_score  
import numpy as np  
  
# 读取数据  
data = np.loadtxt('data\_multivar.txt', delimiter=',')  
  
# 划分特征和标签  
X = data[:, :-1]  
y = data[:, -1]  
  
# 划分训练集和测试集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# 创建线性回归模型  
model = LinearRegression()  
  
# 训练模型  
model.fit(X\_train, y\_train)  
  
# 预测训练集结果  
y\_train\_pred = model.predict(X\_train)  
  
# 计算训练集的均方差MSE  
mse = mean\_squared\_error(y\_train, y\_train\_pred)  
  
# 计算训练集的平均绝对误差MAE  
mae = mean\_absolute\_error(y\_train, y\_train\_pred)  
  
# 计算R方得分  
r2 = r2\_score(y\_train, y\_train\_pred)  
  
# 计算解释方差  
explained\_variance = explained\_variance\_score(y\_train, y\_train\_pred)  
  
# 输出训练集的预测结果  
print("训练集的预测结果：")  
print(y\_train\_pred)  
  
# 输出训练集的均方差MSE  
print("训练集的均方差MSE：", mse)  
  
# 输出训练集的平均绝对误差MAE  
print("训练集的平均绝对误差MAE：", mae)  
  
# 输出R方得分  
print("R方得分：", r2)  
  
# 输出解释方差  
print("解释方差：", explained\_variance)



### 中间的预测数据太多了，我直接略过了不截屏了



1. 使用训练过的模型预测测试集的结果，输出均方差MSE和平均绝对误差MAE

from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score, explained\_variance\_score  
import numpy as np  
  
# 读取数据  
data = np.loadtxt('data\_multivar.txt', delimiter=',')  
  
# 划分特征和标签  
X = data[:, :-1]  
y = data[:, -1]  
  
# 划分训练集和测试集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# 创建线性回归模型  
model = LinearRegression()  
  
# 训练模型  
model.fit(X\_train, y\_train)  
  
# 预测测试集结果  
y\_test\_pred = model.predict(X\_test)  
# 计算测试集的均方差MSE  
mse\_test = mean\_squared\_error(y\_test, y\_test\_pred)  
  
# 计算测试集的平均绝对误差MAE  
mae\_test = mean\_absolute\_error(y\_test, y\_test\_pred)  
# 输出测试集的均方差MSE  
print("测试集的均方差MSE：", mse\_test)  
  
# 输出测试集的平均绝对误差MAE  
print("测试集的平均绝对误差MAE：", mae\_test)

