南华大学计算机学院

**实 验 报 告**

**课程名称 机器学习**

**实验名称 室内用户运动时序数据分类**

**姓 名 李驰浩**

**学 号 20204352119**

**专 业 人工智能**

**班 级 软智01班**

**日 期 2022年6月13日**

南华大学

**1.实验目的**

掌握机器学习的sk-learn。

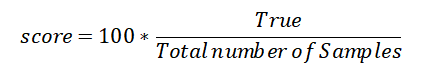
基于python语言，依选用模型进行学习。

**2.实验环境**

Windows 10

Pycharm 2021.1

**3.实验要求**

1、准确率（Accuracy）：对于给定的测试数据集，预测正确的样本数与实际总样本数之比  
2、True，表示预测正确的样本数数量  
3、Total Number of Samples，表示实际总样本数数量  
计算公式如下：  


4.实验设计

（1）数据说明

数据整理自网上公开数据集UCI（已脱敏），数据集涵盖2类不同时间序列，该类数据集广泛应用于时序分类的业务场景。

| **文件类别** | **文件名** | **文件内容** |
| --- | --- | --- |
| 训练集 | train.csv | 训练数据集标签文件，标签CLASS |
| 测试集 | test.csv | 测试数据集标签文件，无标签 |
| 字段说明 | 字段说明.xlsx | 训练集/测试集XXX个字段的具体说明 |
| 提交样例 | Ssample\_submission.csv | 仅有两个字段ID\CLASS |

（2）字段说明

| **名称** | **标签** |
| --- | --- |
| ID | 时间序列标识号 |
| T0 | T0时刻特征 |
| T1 | T1时刻特征 |
| … | … |
| T239 | T239时刻特征 |
| CLASS | 时间序列类别标记 |

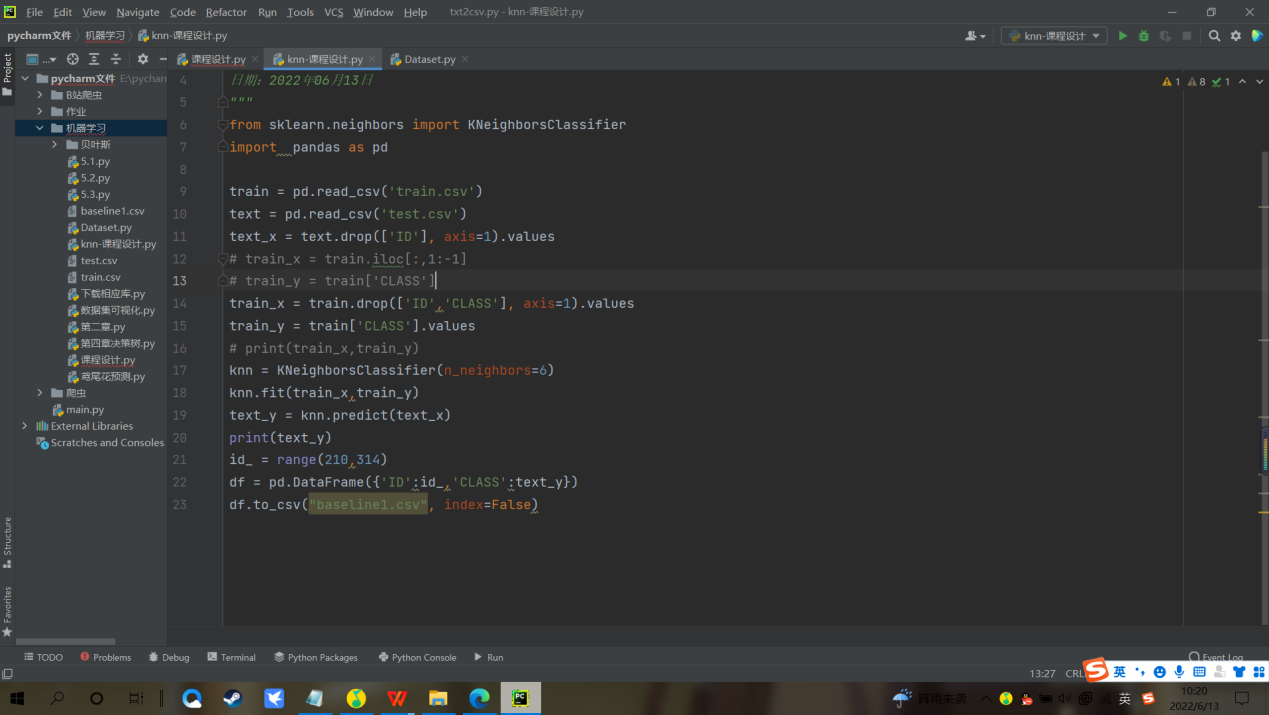
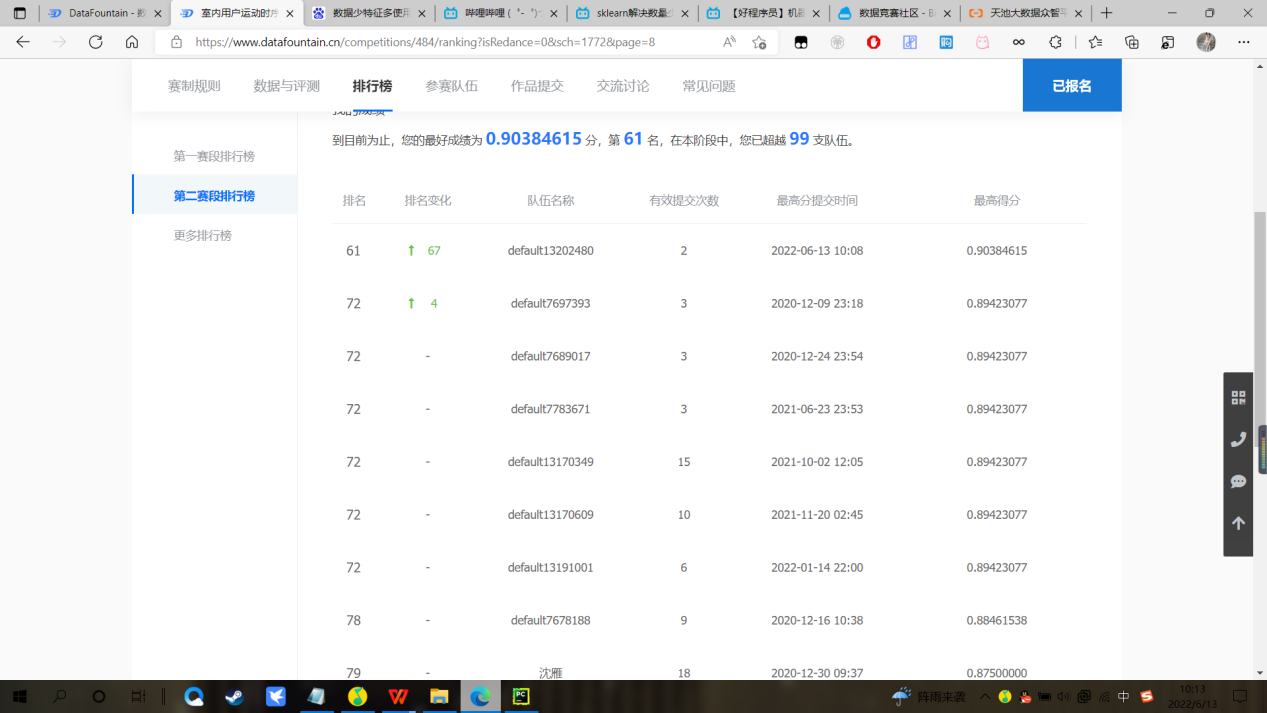
本题是一个二分类的问题，通过对训练集数据的观察，发现数据量很小（210个）且拥有大量的特征（240个），并且对于训练数据的标签值，0和1的分布十分均匀（约各一半）。但是使用KNN算法进行分类，预测的结果并不比支持向量机差。

两个算法在这个实验的结果一样，我猜测可能KNN算法对于多特征的数据有内置的

调整超参数的代码。

5.程序和结果

from sklearn.neighbors import KNeighborsClassifier  
import pandas as pd  
  
train = pd.read\_csv('train.csv')  
text = pd.read\_csv('test.csv')  
text\_x = text.drop(['ID'], axis=1).values  
# train\_x = train.iloc[:,1:-1]  
# train\_y = train['CLASS']  
train\_x = train.drop(['ID','CLASS'], axis=1).values  
train\_y = train['CLASS'].values  
# print(train\_x,train\_y)  
knn = KNeighborsClassifier(n\_neighbors=3)  
knn.fit(train\_x,train\_y)  
text\_y = knn.predict(text\_x)  
print(text\_y)  
id\_ = range(210,314)  
df = pd.DataFrame({'ID':id\_,'CLASS':text\_y})  
df.to\_csv("baseline.csv", index=False)



**6.实验总结**

通过学习和使用sk-learn算法，我对其有了进一步的认识：

scikit-learn是用python实现的机器学习算法库。它具有各种分类，回归和聚类算法，包括支持向量机，随机森林，梯度增强，k-means 和 DBSCAN，旨在与 Python 数值和科学库 NumPy 和 SciPy 互操作。Scikit-learn 主要用 Python 编写，一些用 Cython 编写的核心算法来实现性能。支持向量机由 LIBSVM 周围的 Cython 包装器实现 ; 逻辑回归和线性支持向量机由 LIBLINEAR 周围的类似包装器。

本题具体用法可说：

1、加载数据集，加载文件为train.csv和test.csv：

from sklearn.neighbors import KNeighborsClassifier

import pandas as pd

train = pd.read\_csv('train.csv')

test = pd.read\_csv('test.csv')

2、分离数据集

train = pd.read\_csv('train.csv')

text = pd.read\_csv('test.csv')

text\_x = text.drop(['ID'], axis=1).values

train\_x = train.drop(['ID','CLASS'], axis=1).values

train\_y = train['CLASS'].values

3、模型搭建及训练

knn = KNeighborsClassifier(n\_neighbors=6)

knn.fit(train\_x,train\_y)

text\_y = knn.predict(text\_x)

4、输出结果

id\_ = range(210,314)

df = pd.DataFrame({'ID':id\_,'CLASS':text\_y})

df.to\_csv("baseline1.csv", index=False)