**课后作业二：概率分类法**

任务：使用贝叶斯估计或MLE（最大似然估计），来预测鸢尾花数据集中花的种类。

数据集：鸢尾花数据集是统计学和机器学习中用于分类的经典数据集。该数据集包含了三种不同的鸢尾花：Setosa、Versicolor和Virginica，每种各50个样本。每个样本有四个属性：萼片长度、萼片宽度、花瓣长度和花瓣宽度，所有的测量单位都是厘米。数据集根据4:1的比例划分为训练集和测试集。概率分类法是一种基于概率理论的方法，适合处理此类分类问题。

**实现：**

1.导入必要的库

from sklearn.model\_selection import train\_test\_split #机器学习中非常重要的库，包括一些分类、回归、聚类、降维、模型选择和预处理

from sklearn.neighbors import KNeighborsClassifier

import matplotlib.pyplot as plt #可视化工具

import pandas as pd #用来分析结构化数据

import numpy as np #提供高性能的矩阵运算

import csv #读写文件的库

2.导入训练数据并提取特征值和目标值

"sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width (cm)"为特征值，"species"为目标值

iris\_data = pd.read\_csv(r'D:\dataenclorse\second\iris\_train.csv')

X = iris\_data[["sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width (cm)"]].values

y = iris\_data["species"].values

3.划分数据

# 构造训练数据和测试数据

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=0)

4.构造knn模型并训练该模型

# 构造KNN模型

knn = KNeighborsClassifier(n\_neighbors=1)

# 训练模型

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

y\_pred = knn.predict(X\_test)

5.加载iris\_test.csv的数据并对iris\_test.csv进行预测

# 做出预测

file\_path = ("D:\dataenclorse\second\iris\_test.csv")

data = pd.read\_csv(r'D:\dataenclorse\second\iris\_test.csv', usecols=["sepal length (cm)","sepal width (cm)","petal length (cm)","petal width (cm)"])

grouped\_data = [data.iloc[i:i+1].values.tolist() for i in range(0, len(data), 1)]

grouped\_data = [sum(sublist, []) for sublist in grouped\_data]

X\_new = np.array(grouped\_data)

prediction = knn.predict(X\_new)

print("预测的目标类别是：{}".format(prediction))

file\_path=r'D:\dataenclorse\second\test.csv'

6.预测结果与加载的数据一起保存到test.csv文件中

getdata和getdata2函数与作业一中一模一样，仅仅有读取文件的内容不同

def getdata(path):

    data\_frame = pd.read\_csv(r'D:\dataenclorse\second\iris\_test.csv')  # skiprows=14

    data\_x,data\_y = np.array(data\_frame['sepal length (cm)']), np.array(data\_frame['sepal width (cm)'])

    return data\_x,data\_y

def getdata2(path):

    data\_frame = pd.read\_csv(r'D:\dataenclorse\second\iris\_test.csv')  # skiprows=14

    data\_p,data\_q= np.array([data\_frame['petal length (cm)'], np.array(data\_frame['petal width (cm)'])])

    return  data\_p,data\_q

data\_x,data\_y=getdata('iris\_test.csv')

data\_p,data\_q=getdata('iris\_test.csv')

with open(file\_path,'w',encoding='utf-8',newline='') as f:

        fieldnames=['sepal length(cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)','class']

        f\_csv = csv.DictWriter(f, fieldnames=fieldnames)

        f\_csv.writeheader()

        for i in range(0,len(prediction)):

            f\_csv.writerow({'sepal length(cm)':data\_x[i],'sepal width (cm)':data\_y[i],'petal length (cm)':data\_p[i],'petal width (cm)':data\_q[i],'class':prediction[i]})

1. 结果如下：

