# -\*- coding: utf-8 -\*-

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.patches as mpatches

import matplotlib as mpl

# 设置属性防止中文乱码

mpl.rcParams['font.sans-serif'] = [u'SimHei']

mpl.rcParams['axes.unicode\_minus'] = False

# 加载鸢尾花数据集

iris = load\_iris()

# 只使用花瓣长度和宽度作为特征（便于可视化)

x, y = iris.data[:, 2:4], iris.target

# 按照标签的比例分成训练集和测试集

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, stratify=y, test\_size=0.2, random\_state=42)

# 使用训练集训练一个贝叶斯分类器

gnb = GaussianNB()

gnb.fit(x\_train, y\_train)

# 计算预测值并计算准确率

y\_train\_pred = gnb.predict(x\_train)

train\_acc = accuracy\_score(y\_train, y\_train\_pred)

print(f'训练集准确率: {train\_acc:.4f}')

y\_test\_pred = gnb.predict(x\_test)

test\_acc = accuracy\_score(y\_test, y\_test\_pred)

print(f'测试集准确率: {test\_acc:.4f}')

# 将分类器绘制到图中

x1\_min, x1\_max = min(x[:, 0]) - 1.0, max(x[:, 0]) + 1.0

x2\_min, x2\_max = min(x[:, 1]) - 1.0, max(x[:, 1]) + 1.0

step\_size = 0.01

x1\_values, x2\_values = np.meshgrid(np.arange(x1\_min, x1\_max, step\_size),

np.arange(x2\_min, x2\_max, step\_size))

cm\_light = mpl.colors.ListedColormap(['#A0FFA0', '#FFA0A0', '#A0A0FF'])

cm\_dark = mpl.colors.ListedColormap(['g', 'r', 'b'])

mesh\_output = gnb.predict(np.c\_[x1\_values.ravel(), x2\_values.ravel()])

mesh\_output = mesh\_output.reshape(x1\_values.shape)

plt.figure()

plt.pcolormesh(x1\_values, x2\_values, mesh\_output, cmap=cm\_light)

plt.scatter(x[:, 0], x[:, 1], c=y, s=80, edgecolors='black', linewidth=1, cmap=cm\_dark)

plt.legend(handles=[mpatches.Patch(color='g', label='山鸢尾'),

mpatches.Patch(color='r', label='变色鸢尾'),

mpatches.Patch(color='b', label='维吉尼亚鸢尾')])

plt.xlim(x1\_values.min(), x1\_values.max())

plt.ylim(x2\_values.min(), x2\_values.max())

plt.xticks((np.arange(int(x1\_min), int(x1\_max)+1, 1.0)))

plt.yticks((np.arange(int(x2\_min), int(x2\_max)+1, 1.0)))

plt.xlabel('花瓣长度/cm')

plt.ylabel('花瓣宽度/cm')

plt.title('朴素贝叶斯鸢尾花分类器')

plt.show()