

```

In [4]: # 导入操作系统库
import os
# 更改工作目录
os.chdir(r"D:\softwares\applied statistics\pythoncodelearning\chap5\sourcecode")
# 导入基础计算库
import numpy as np
# 导入绘图库
import matplotlib.pyplot as plt
# 导入数据生成工具
from sklearn.datasets import load_iris
# 导入决策树分类器
from sklearn.tree import DecisionTreeClassifier
# 导入决策边界显示工具
from sklearn.inspection import DecisionBoundaryDisplay
# 导入绘图库中的字体管理包
from matplotlib import font_manager
# 实现中文字符正常显示
font = font_manager.FontProperties(fname=r"C:\Windows\Fonts\SimKai.ttf")
# 使用seaborn风格绘图
plt.style.use("seaborn-v0_8")
# 获取数据
iris = load_iris()
# 类别数
n_classes = 3
# 绘图颜色
plot_colors = "ryb"
# 步长
plot_step = 0.02
fig, axs = plt.subplots(2, 3, figsize=(6,6), tight_layout=True)
for pairidx, pair in enumerate([[0, 1], [0, 2], [0, 3], [1, 2], [1, 3], [2, 3]]):
    # 两个维度
    X = iris.data[:, pair]
    y = iris.target
    # 决策树建模
    clf = DecisionTreeClassifier()
    # 模型拟合
    clf.fit(X, y)
    # 绘制决策边界
    ax = axs.flatten()[pairidx]
    DecisionBoundaryDisplay.from_estimator(
        clf,
        X,
        cmap=plt.cm.RdYlBu,
        response_method="predict",
        ax=ax,
        xlabel=iris.feature_names[pair[0]],
        ylabel=iris.feature_names[pair[1]],
    )
# 训练样本散点图
for i, color in zip(range(n_classes), plot_colors):
    idx = np.where(y == i)
    ax.scatter(
        X[idx, 0],
        X[idx, 1],
        c=color,
        label=iris.target_names[i],
        edgecolor="black",
        s=15,
    )

```

```

)

plt.suptitle("Decision surface of decision trees trained on pairs of features")
ax.legend(loc="lower right", borderpad=0, handletextpad=0)
plt.show()
fig.savefig("../codeimage/code1.pdf")

```

