基于决策树模型对 IRIS 数据集分类

1 python 实现

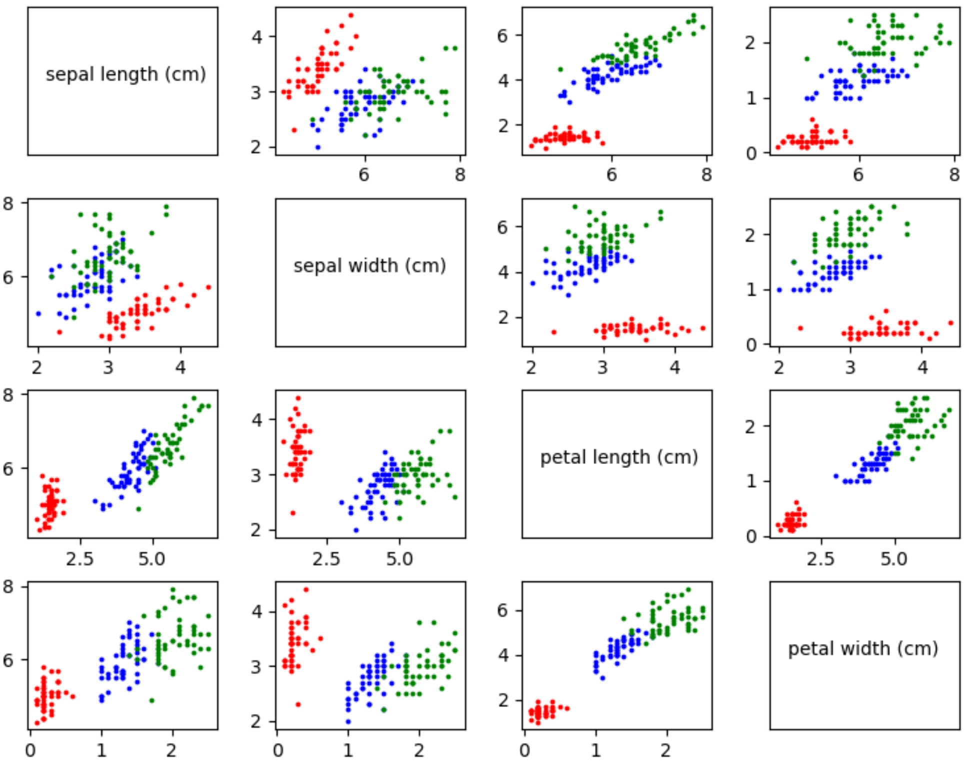
加载数据集

IRIS 数据集在 sklearn 模块中已经提供。

```
# -*- coding: utf-8 -*-

from matplotlib import pyplot as plt
import numpy as np
from sklearn import tree
from sklearn.datasets import load_iris

if __name__ == '__main__':
    print('\n\n\n\n\n\n\n\n\n\n')

    # show data info
    data = load_iris()
    print('keys: \n', data.keys()) # ['data', 'target', 'target_names', 'DESCR',
'feature_names']
    feature_names = data.get('feature_names')
    print('feature names: \n', data.get('feature_names'))
    print('target names: \n', data.get('target_names'))
    x = data.get('data')
    y = data.get('target')
    print(x.shape, y.shape)
    print(x)
    print(data.get('DESCR'))
```

可视化数据集

```
# visualize the data
    f = []
    f.append(y==0)
    f.append(y==1)
    f.append(y==2)
    color = ['red','blue','green']
    fig, axes = plt.subplots(4,4)
    for i, ax in enumerate(axes.flat):
        row  = i // 4
        col = i % 4
        if row == col:
            ax.text(.1,.5, feature_names[row])
            ax.set_xticks([])
            ax.set_yticks([])
        continue
    for  k in range(3):
        ax.scatter(x[f[k],row], x[f[k],col], c=color[k], s=3)
fig.subplots_adjust(hspace=0.3, wspace=0.3) # 设置间距
plt.show()
```



分类和预测

```
# 划分训练集和测试集
num = x.shape[0]
ratio = 7/3 # 训练集数目: 测试集数目
num_test = int(num/(1+ratio))
num_train = num -  num_test
index = np.arange(num)
np.random.shuffle(index)
x_test = x[index[:num_test],:]
y_test = y[index[:num_test]]
x_train = x[index[num_test:],:]
y_train = y[index[num_test:]]

# 构建决策树
clf = tree.DecisionTreeClassifier()
clf.fit(x_train, y_train)

# 预测
```

```
    y_test_pre = clf.predict(x_test)
    print('the predict values are', y_test_pre)
```

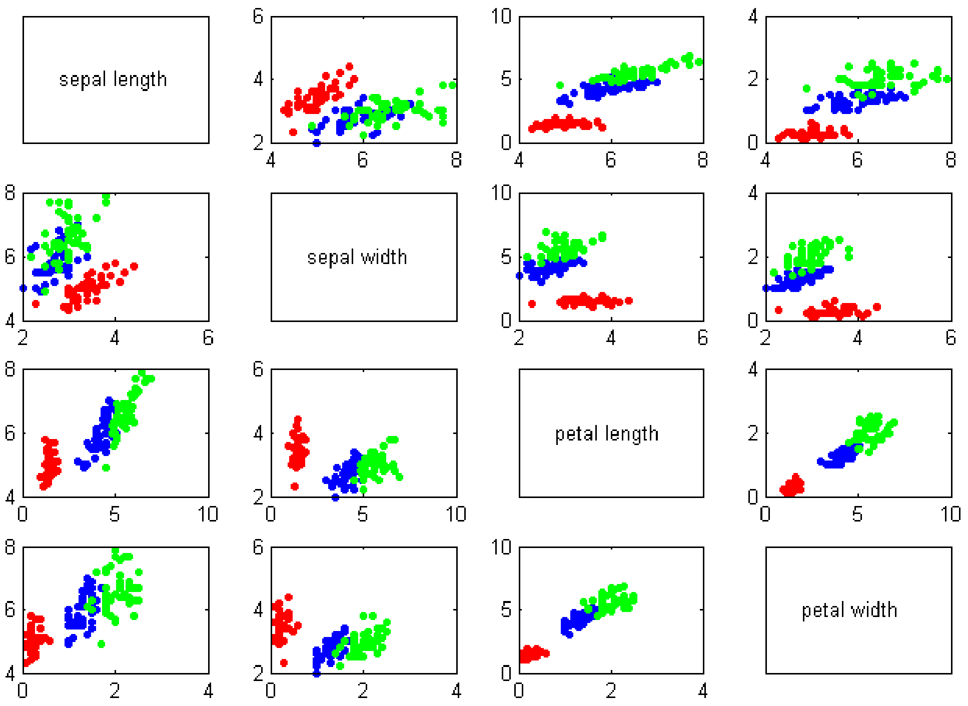
计算准确率

```
  # 计算分类准确率
    acc = sum(y_test_pre==y_test)/num_test
    print('the accuracy is', acc)
```

由于数据集的划分是随机的每次得到的准确率都不一样，一般位于91%-97%之间。

2 基于MATLAB 实现

Matlab 对数据的可视化



实现代码如下：

```
clc
clear all
close all;
load fisheriris;
x = meas;
y = species;
class = unique(y);
attr = {'sepal length', 'sepal width', 'petal length', 'petal width'};
ind1 = ismember(y, class{1});
ind2 = ismember(y, class{2});
ind3 = ismember(y, class{3});
s=10;
```

```
for i=1:4
   for j=1:4
      subplot(4,4,4*(i-1)+j);
      if i==j
          set(gca, 'xtick', [], 'ytick', []);
          text(.2, .5, attr{i});
          set(gca, 'box', 'on');
          continue;
       end
      scatter(x(ind1,i), x(ind1,j), s, 'r', 'MarkerFaceColor', 'r');
      hold on
      scatter(x(ind2,i), x(ind2,j), s, 'b', 'MarkerFaceColor', 'b');
      hold on
      scatter(x(ind3,i), x(ind3,j), s, 'g', 'MarkerFaceColor', 'g');
       set(gca, 'box', 'on');
   end
end

ratio = 7/3;
num = length(x);
num_test = round(num/(1+ratio));
num_train = num - num_test;
index = randperm(num);
x_train = x(index(1:num_train),:);
y_train = y(index(1:num_train));
x_test = x(index(num_train+1:end),:);
y_test = y(index(num_train+1:end));

tree = fitctree(x_train, y_train);
y_test_p = predict(tree, x_test);
acc = sum(strcmp(y_test,y_test_p))/num_test;
disp(['The accuracy is ', num2str(acc)]);
```