实验三　支持向量机

实验内容：支持向量机

20201060287

李昂

环境要求：

Python，numpy支持多维度的数组和矩阵运算，pandas数据处理和分析工具，Matplotlib图形化工具，sklearn机器学习库

```
import pandas as pd
import sklearn.ensemble as ensemble
from sklearn.model_selection import cross_val_score, GridSearchCV,train_test_split
import sklearn.tree
import pydotplus
import pprint
```

任务一：线性可分类问题

题目：

本任务中你将使用机器学习库sklearn中的支持向量机的线性核函数来进行线性可分类数据的决策边界的构建，将结果可视化，并尝试不同的惩罚系数C，观察对结果的影响并分析。文件ex3data1.csv包含我们的线性可分类问题的数据集。x1，y1分别代表纵横坐标，a代表标签。

请将70%的数据用作训练集，30%的数据用作测试集，使用留出法对以上模型进行验证。

代码：

```
# -*- coding: utf-8 -*-
# @Time : 4/3/23 16:12
# @Author : ANG


import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn import svm
from sklearn.model_selection import train_test_split


def load_data(path):
    """
    数据预处理
    :return: 特征数据和标签数据
    """
    data = pd.read_csv(path)
    # 将第一列和第二列的数据作为特征
    x = data.loc[:, ['x1', 'y1']].values
    # 将第三列的数据作为标签
    label = data.loc[:, ['a']].values
```

```
    # 将数据集分为训练集和测试集
    x_train, x_test, label_train, label_test = train_test_split(x, label, test_size=0.3, random_state=0)

    return x_train, x_test, label_train, label_test


x_train, x_test, label_train, label_test = load_data(
    '/Users/wallanceleon/Desktop/机器学习/机器学习实验/20201060287-李昂-实验三/ex3data1.csv')


def SVM_linear_kernal(x_train, x_test, label_train, label_test):
    """
    SVM线性核函数
    :param x_train: 训练集特征
    :param x_test: 测试集特征
    :param label_train: 训练集标签
    :param label_test: 测试集标签
    :return: None
    """
    # 创建SVM分类器
    clf = svm.SVC(kernel='linear')

    # 尝试不同的惩罚系数C进行训练和测试
    for C in [0.1, 1, 10]:
        # 设定惩罚系数C
        clf.set_params(C=C)
        # 训练
        clf.fit(x_train, label_train)
        # 预测
        y_pred = clf.predict(x_test)
        # 评估模型性能
        train_score = clf.score(x_train, label_train)
        test_score = clf.score(x_test, label_test)
        print(f"C={C}, train score={train_score:.3f}, test score={test_score:.3f}")
        # 依据训练集画出决策边界
        plt.figure()
        plt.title(f'C={C}')
        # 限定x轴和y轴的范围
        plt.xlim(0, 5)
        plt.ylim(1, 5)
        plt.scatter(x_train[:, 0], x_train[:, 1], c=label_train, s=20, cmap=plt.cm.Paired)
        plt.xlabel('x1')
        plt.ylabel('y1')
        x_min, x_max = x_train[:, 0].min() - 1, x_train[:, 0].max() + 1
        y_min, y_max = x_train[:, 1].min() - 1, x_train[:, 1].max() + 1
        xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02), np.arange(y_min, y_max, 0.02))
        z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
        z = z.reshape(xx.shape)
        plt.contour(xx, yy, z, colors='k', levels=[-1, 0, 1], alpha=0.5, linestyles=['--', '-', '--'])
        plt.show()

        # 画出不同参数对测试集的分裂效果
        plt.figure()
        plt.title(f'C={C}')
        # 限定x轴和y轴的范围
        plt.xlim(0, 5)
        plt.ylim(1, 5)
        plt.scatter(x_test[:, 0], x_test[:, 1], c=label_test, s=20, cmap=plt.cm.Paired)
        plt.xlabel('x1')
        plt.ylabel('y1')
        # 画出决策边界
        x_min, x_max = x_train[:, 0].min() - 1, x_train[:, 0].max() + 1
        y_min, y_max = x_train[:, 1].min() - 1, x_train[:, 1].max() + 1
        xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02), np.arange(y_min, y_max, 0.02))
        z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
        z = z.reshape(xx.shape)
        plt.contour(xx, yy, z, colors='k', levels=[-1, 0, 1], alpha=0.5, linestyles=['--', '-', '--'])
        plt.show()
```

```
SVM_linear_kernal(x_train, x_test, label_train, label_test)
```

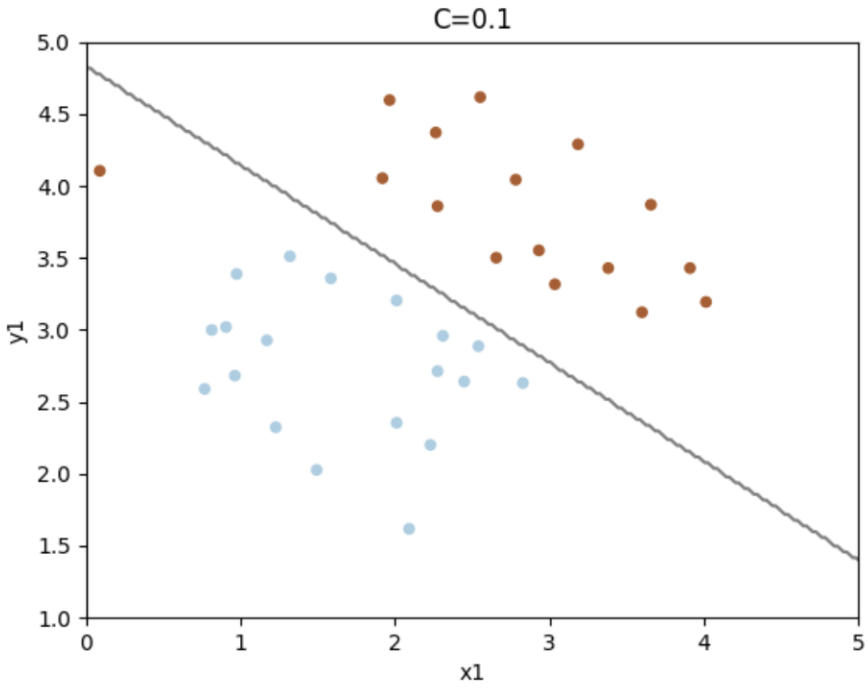
分析：

支持向量机的线性核函数可以很好地解决线性可分问题。通过调整惩罚系数C，验证训练集和测试集的准确率。

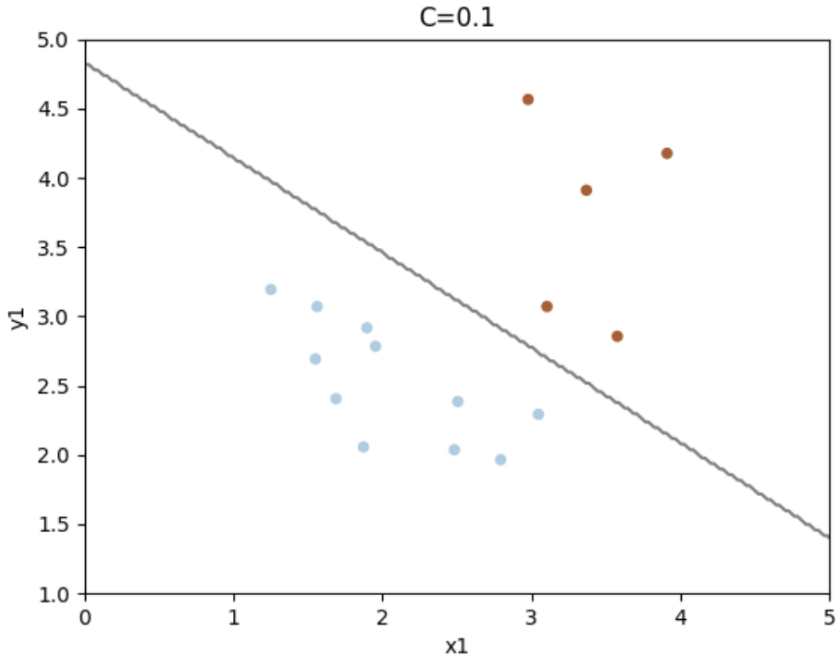
整体而言，本次实验难度不大，基本思路同实验一、二差异不大，都是对数据进行预处理，划分测试和训练集，最后调用sklearn库函数进行验证，需要了解的更多的是函数的参数值的选择和图表可视化

实验结果

> c=0.1

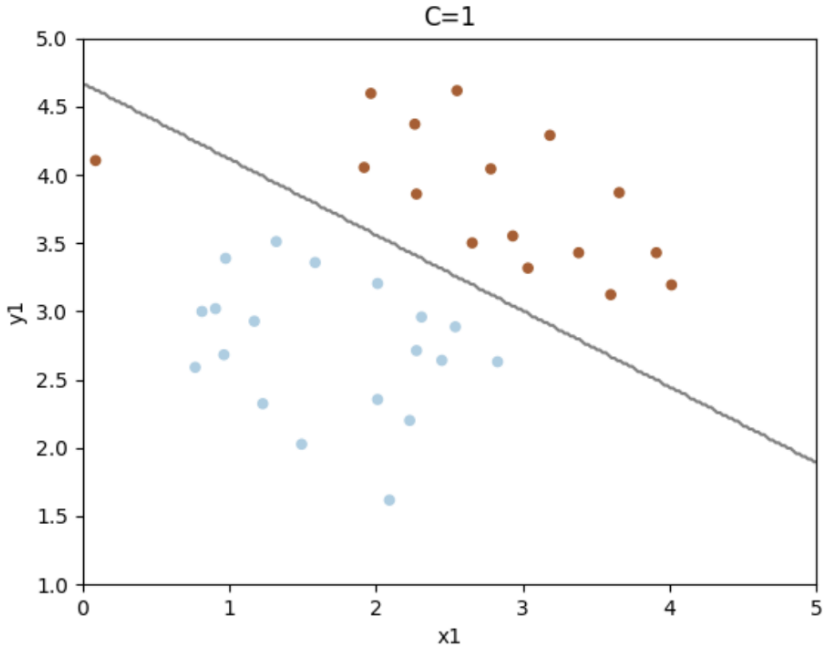


训练集

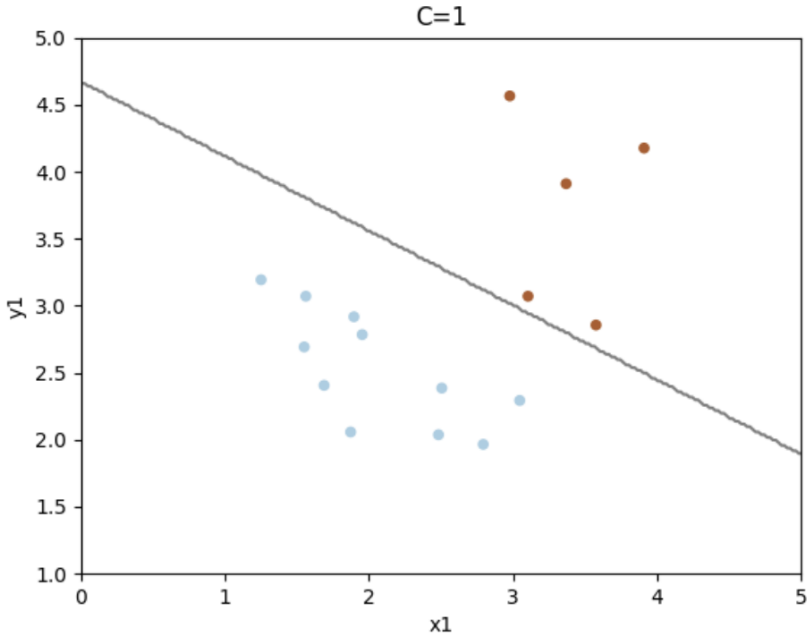

C=0.1
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
y1
0
1
2
3
4
5
x1

测试集

c=1

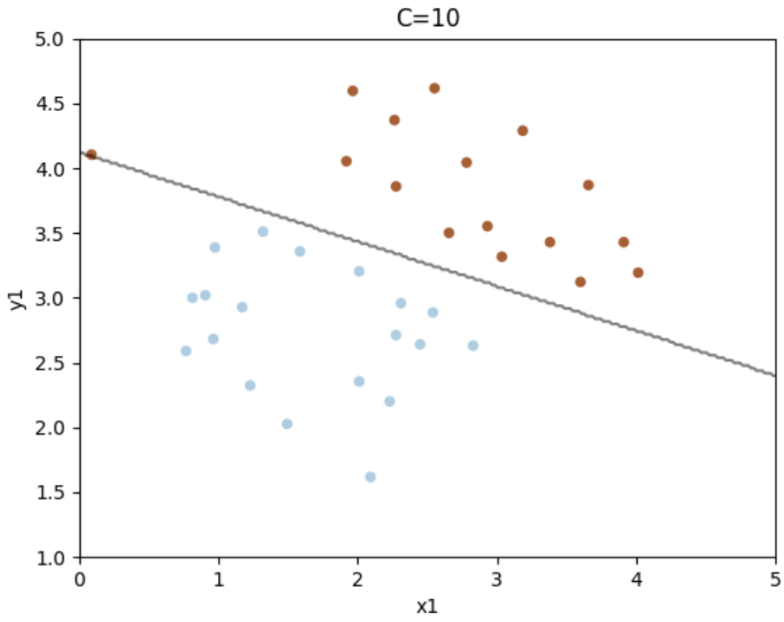

C=1
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
y1
0
1
2
3
4
5
x1

训练集

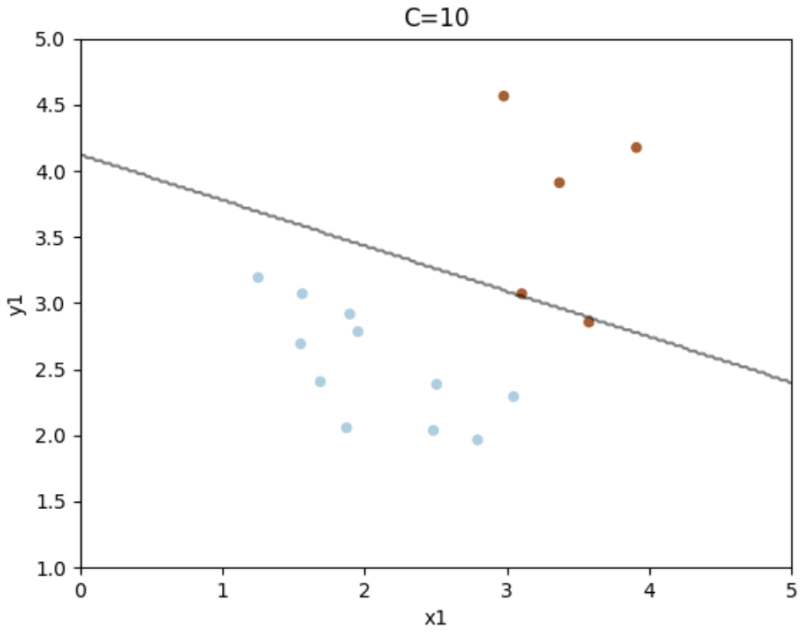

C=1
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
y1
0
1
2
3
4
5
x1

测试集

> c=10


C=10
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
y1
0
1
2
3
4
5
x1

训练集



测试集

任务二：线性不可分问题

题目：

本任务中你将使用机器学习库sklearn中支持向量机的高斯核函数来进行非线性数据的支持向量的构建，将结果可视化，尝试不同的核函数系数gamma，观察对结果的影响并分析。文件ex3data2.csv包含我们的线性不可分问题的数据集。x1，y1分别代表纵横坐标，a代表标签。

请将70%的数据用作训练集，30%的数据用作测试集，使用留出法对以上模型进行验证。

代码：

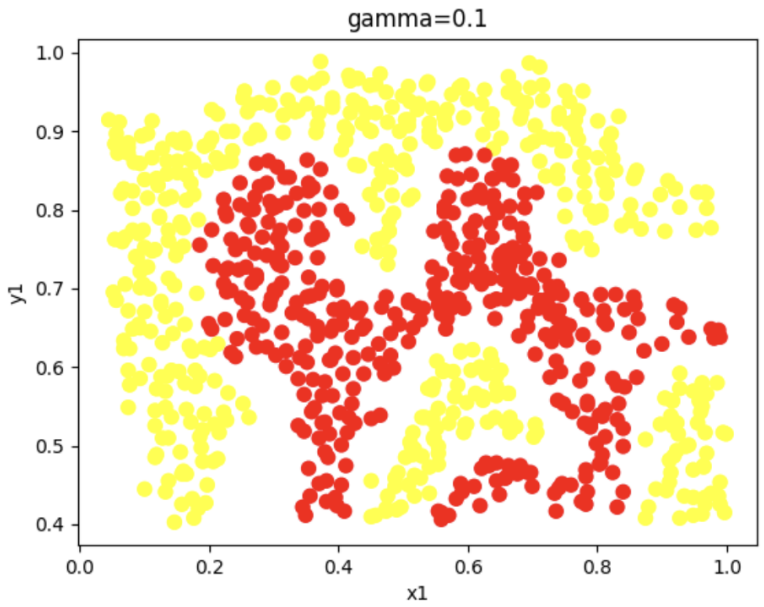
```
# -*- coding: utf-8 -*-
# @Time : 4/3/23 17:34
# @Author : ANG

import pandas as pd
from sklearn.svm import SVC
from matplotlib import pyplot as plt

# 读入数据
data = pd.read_csv('/Users/wallanceleon/Desktop/机器学习/机器学习实验/20201060287-李昂-实验三/ex3data2.csv')

# 构建特征和标签
X = data[['x1', 'y1']]
y = data['a']

# 构建支持向量机，使用高斯核函数
for gamma in [0.1, 1, 10, 100]:
    svc = SVC(kernel='rbf', gamma=gamma).fit(X, y)
```

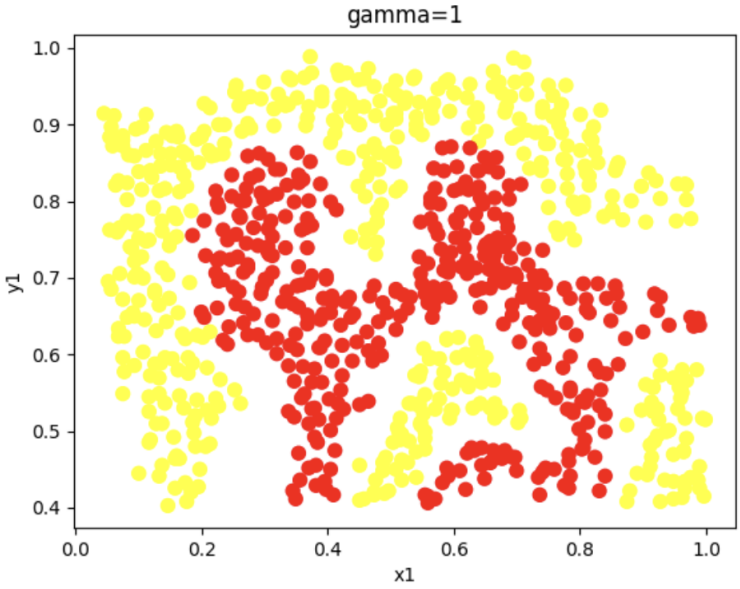
```
    # 预测并输出准确率
    accuracy = svc.score(X, y)
    print(f"gamma={gamma}, Accuracy={accuracy:.3f}")

    # 可视化
    plt.figure()
    plt.title(f"gamma={gamma}")
    plt.scatter(X['x1'], X['y1'], c=y, s=50, cmap='autumn')
    plt.xlabel('x1')
    plt.ylabel('y1')
    plt.show()
```

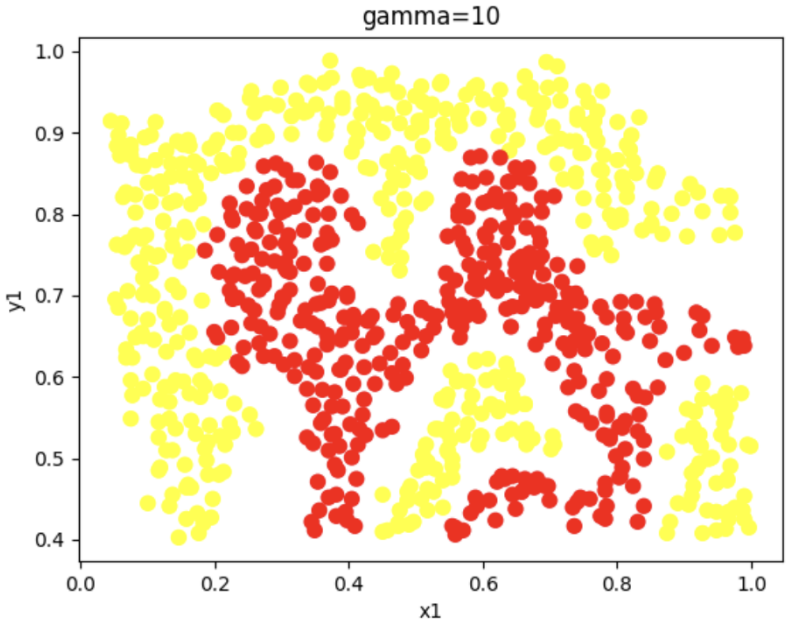
分析：

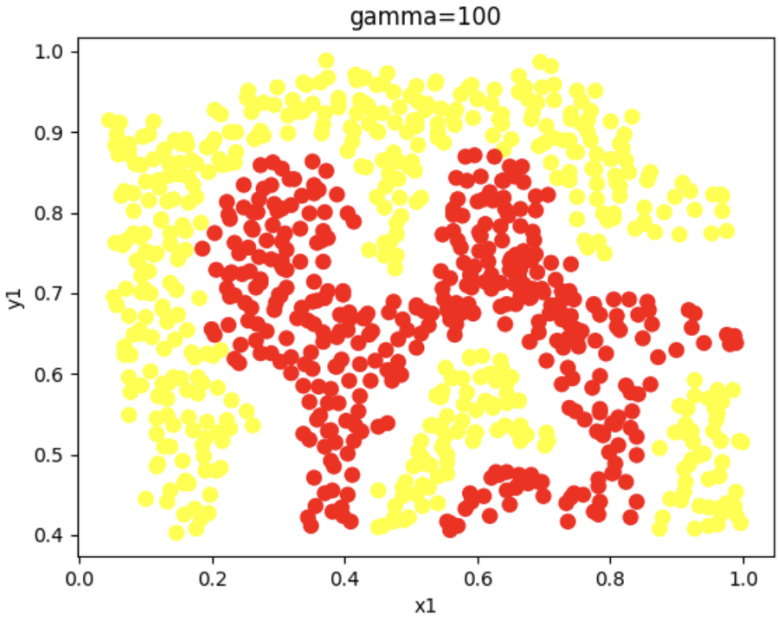
本题思路与上一题目大同小异，只不过是改变了需要调用的函数，换汤不换药，这里不再赘述

实验结果




gamma=1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
y1
0.0
0.2
0.4
0.6
0.8
1.0
x1


gamma=10
1.0
0.9
0.8
0.7
0.6
0.5
0.4
y1
0.0
0.2
0.4
0.6
0.8
1.0
x1


gamma=100
1.0
0.9
0.8
0.7
0.6
0.5
0.4
y1
0.0
0.2
0.4
0.6
0.8
1.0
x1