**机器学习报告名称：支持向量机（二分类）的实现和应用**

**一、基本信息**

**姓名:**刘俊

**学号：**41821153

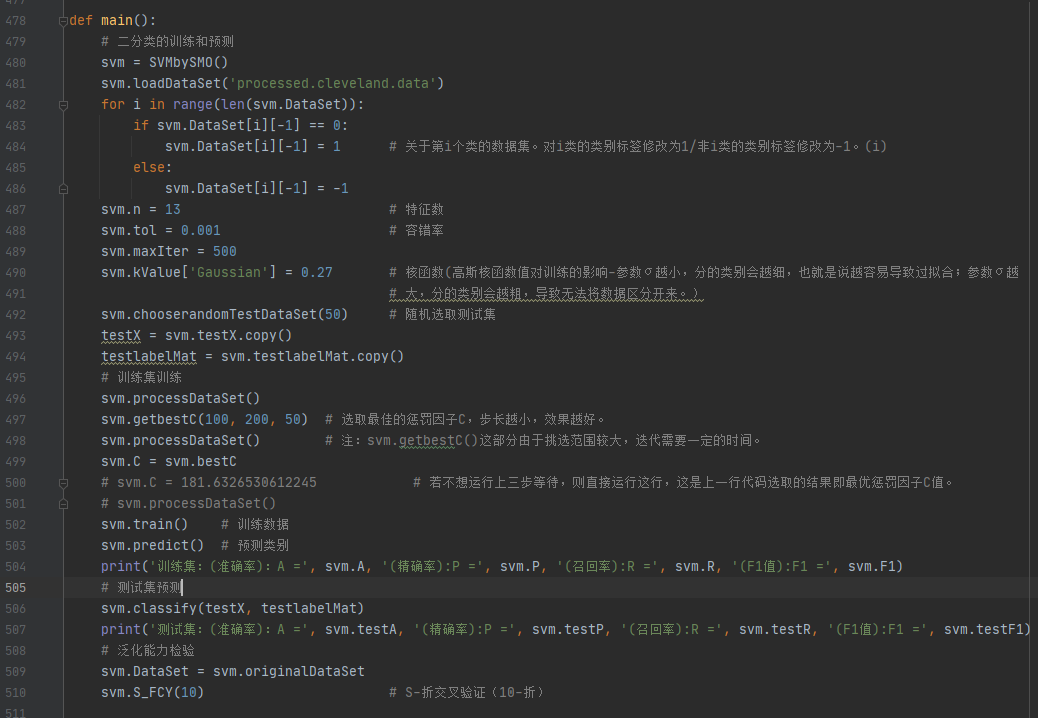
**二、上机作业内容与结果（包括图形截图）：**

1目的：实现二分类支持向量机算法（SMO算法）的代码编写，并用其对心脏病数据（processed.cleveland.data）进行训练和分类预测。

2 结果：

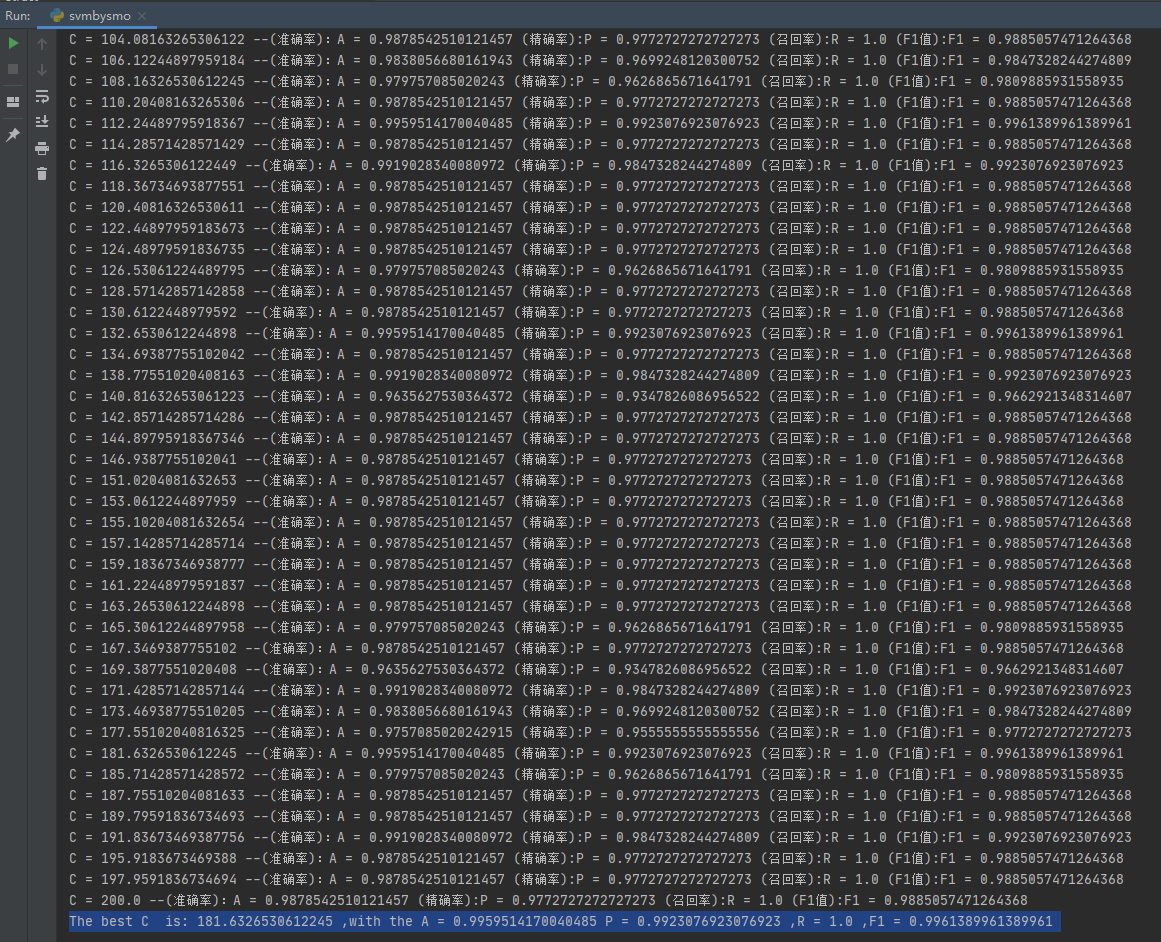
泛化能力：0.915（见最后两图）

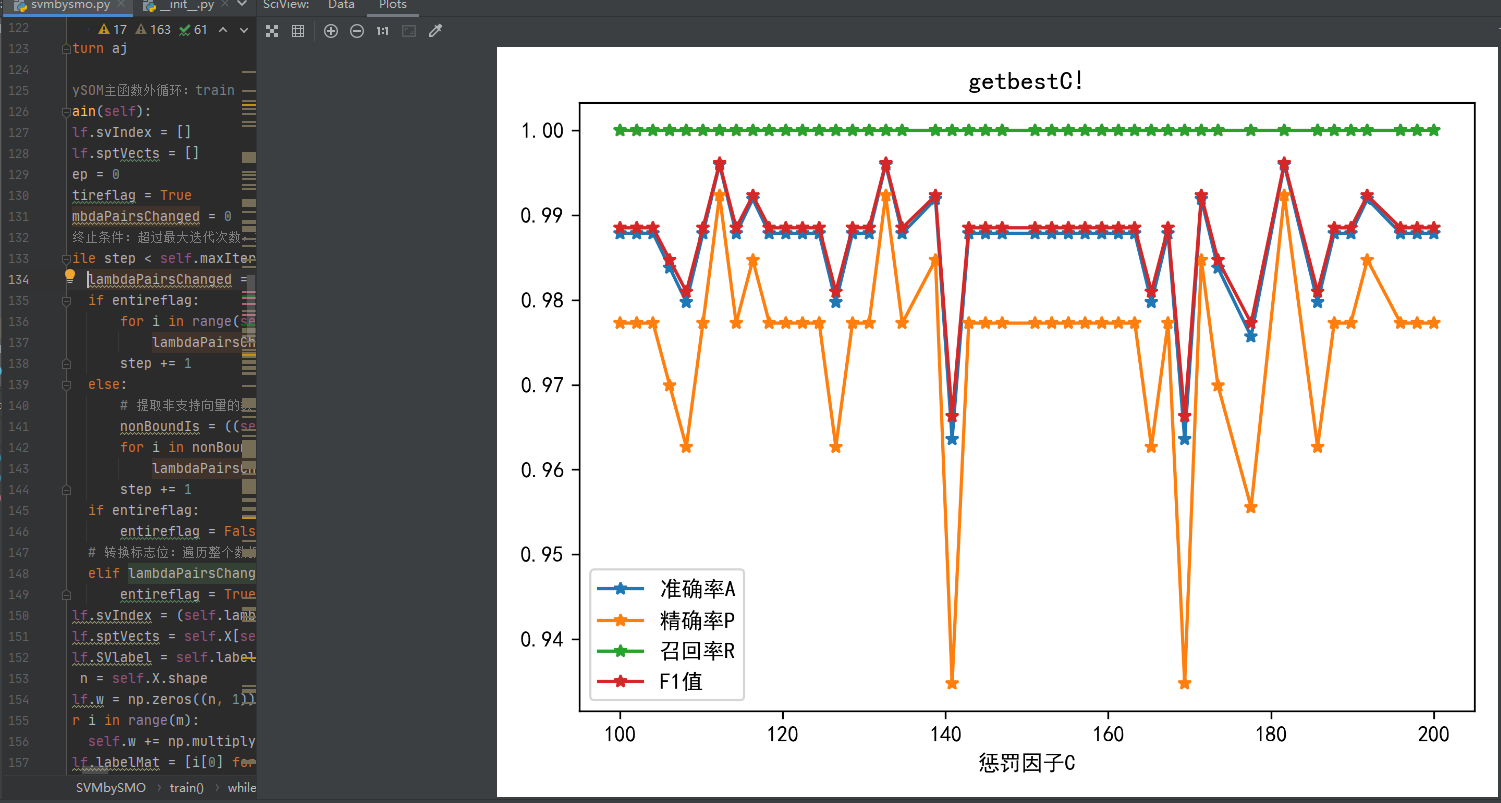
**main()函数：**



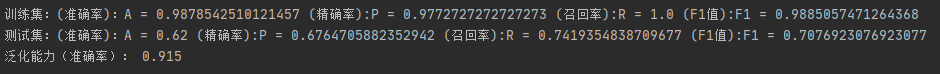
**其中：**

svm.C = svm.getbestC(100, 200, 50)

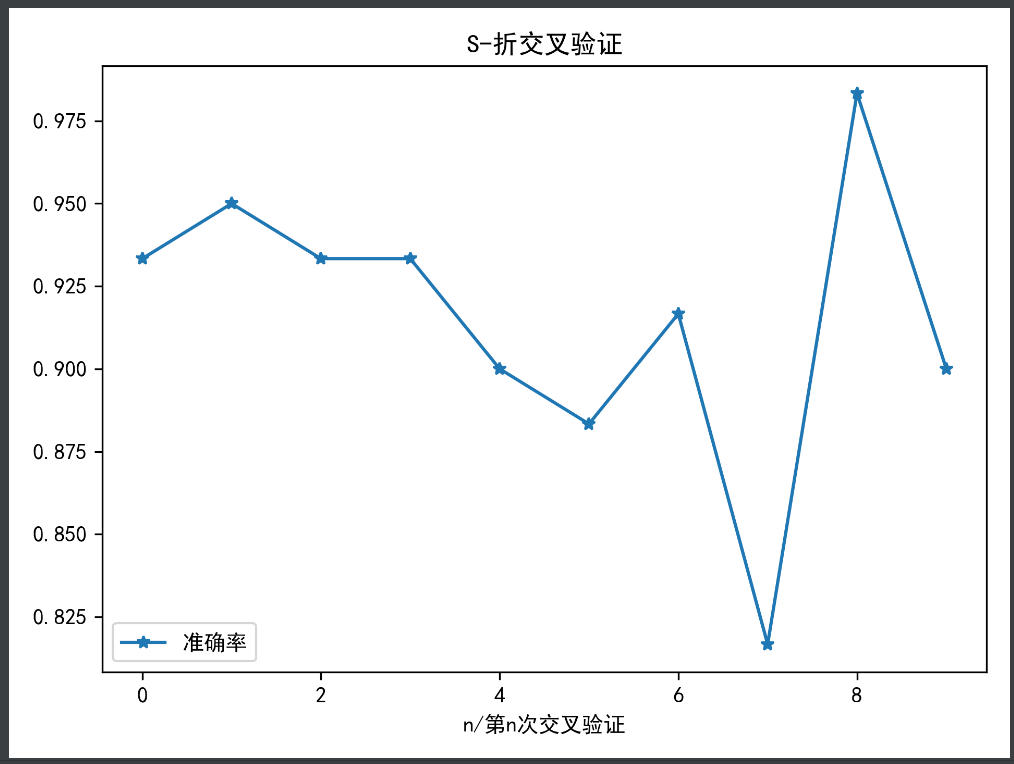




# 训练和预测：



svm.S\_FCY(10) # S-折交叉验证（10-折）



**三、文件与代码说明：**

<机器学习课程-code>

**SVMbySMO.py:**主要包括SVMbySMO类和MultiSVMbySMO(SVMbySMO)类。

其中类的方法主要包括：

# 加载数据集，并将数据储存为列表结构。  
def loadDataSet(self, filename):

# 通过数据集初始化参数  
def initparam(self):

# 核函数  
def kernels(self, dataMat, A):

# 选择lambda2,从缓存中寻找符合KKT条件并具有最大误差的j  
def chooseJ(self, i, Ei):

# SVMbySOM主函数外循环：train  
def train(self):

# SVMbySOM主函数内循环：innerLoop  
def innerLoop(self, i):

# 训练数据集的预测  
def predict(self):

# 二类分类器  
def classify(self, testSet, testLabel): # 测试集testSet及其分类标签testLabel

# 随机选取50个数据作为测试集。  
def chooserandomTestDataSet(self):

# 获取最佳惩罚因子C的值  
def getbestC(self, min, max, d):

# S-折交叉验证  
def S\_FCY(self, n):

**processed.cleveland.data**：数据集

**四、备注：**

代码均自己编写。

对于数据集，用直接删去的方式简单的处理了缺省值或异常值，并用标准化的数据进行训练和预测。

为了更好的效果，对惩罚因子C取值的选择进行了优化，选取了效果更好的惩罚因子C（181.6326530612245）的取值，而不是随便取的。

测试集由从数据集随机挑选的50个数据组成。

选取了准确率、精确率、召回率、F1值-4个指标。

**高斯核函数参数σ值对训练的影响**-参数σ越小，分的类别会越细，也就是说越容易导致过拟合；参数σ越大，分的类别会越粗，导致无法将数据区分开来。这里进行了调试选择和综合考虑，选择了参数σ=0.27

做了S-折交叉验证检验模型的泛化能力。

此外，自己也试着编写了多分类的SVM的代码，注释中有编写所基于的原理，在多分类参数优化过程中有一定的运算量，借用了百度云高级算力卡的算力完整的过程大概需要10多分钟。