**Homework4 SVM Classifier 实验报告**

**实验要求：**

有2类二维空间点，A类和B类。

A类点以（0，0）为中心、（1，0；0，1）为协方差矩阵的二维高斯分布；

B类点以（1，2）为中心、（1，0；0，2）为协方差矩阵的二维高斯分布；

随机生成300个A类点，200个B类点，并用SVM（可以调用机器学习库）进行分类（选择合适的kernel）。

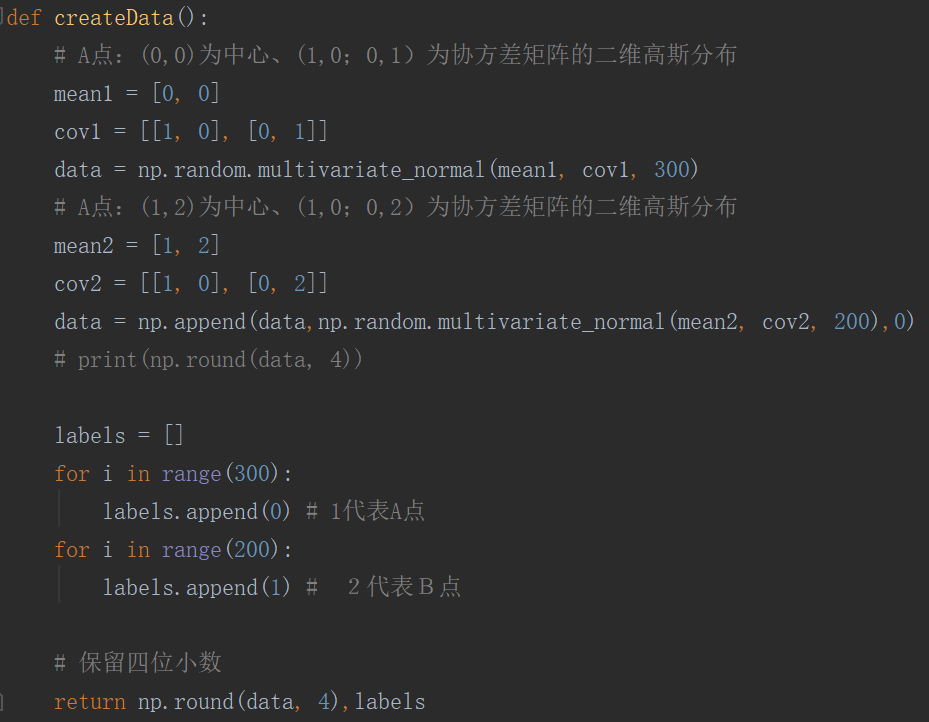
请解释你的SVM调用方法，描述结果。

**实验思路：**

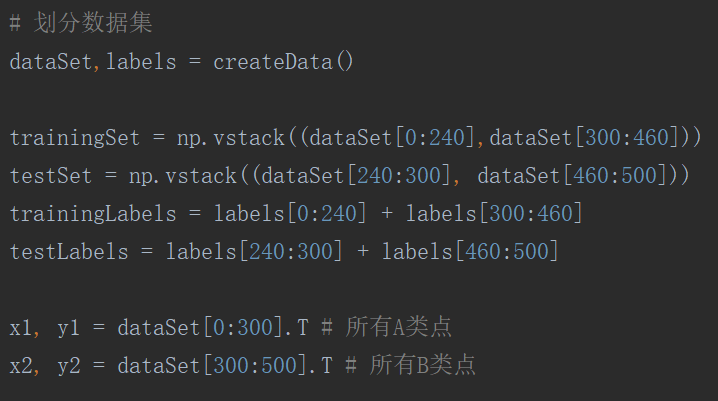
1. 生成数据集：首先生成这500个点，以列表的形式表示位置坐标，再把所有的坐标放入一个大的列表中。再生成一个label列表，包含300个’A’和200个’B’。并以4:1划分训练集和测试集。
2. 生成SVM：用SVC函数生成对应的SVM，选择线性核函数和高斯核函数两种核函数，并用GridSearchCV函数进行调参，选出最佳参数，用到最终的SVM模型中，并用生成的模型对训练集中的数据点进行拟合。
3. 计算准确率。遍历测试集中的每一个点，用svc.predict函数预测结果，与原本的label比较看是否预测正确。
4. 绘图：
   1. AB点分布图。把生成的300个A类点和200个B类点分别用两种颜色的符号绘制在一个二维坐标系内。
   2. 用线性核的SVM得到的超平面图。
   3. 用高斯核的SVM得到的超平面图。

**核心代码片段：**

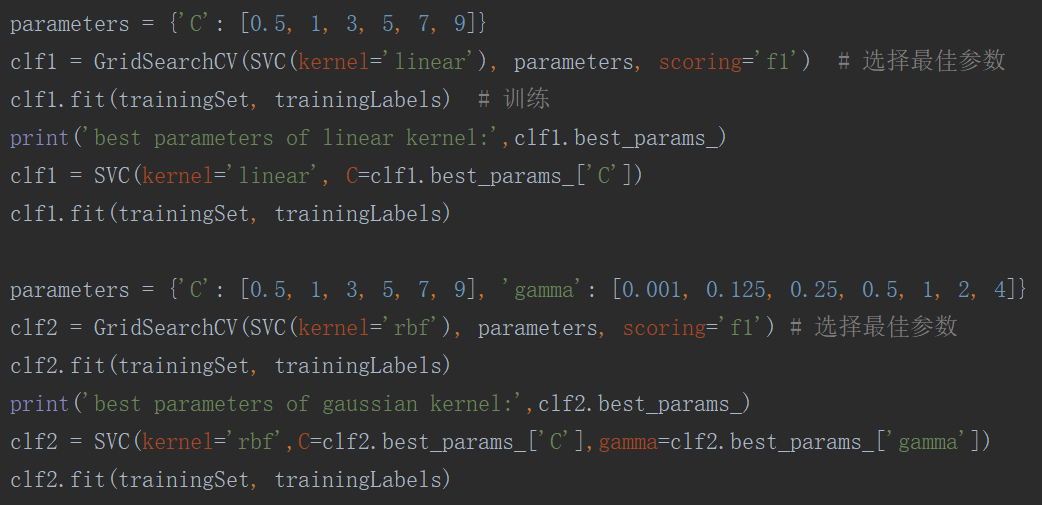
生成数据集：



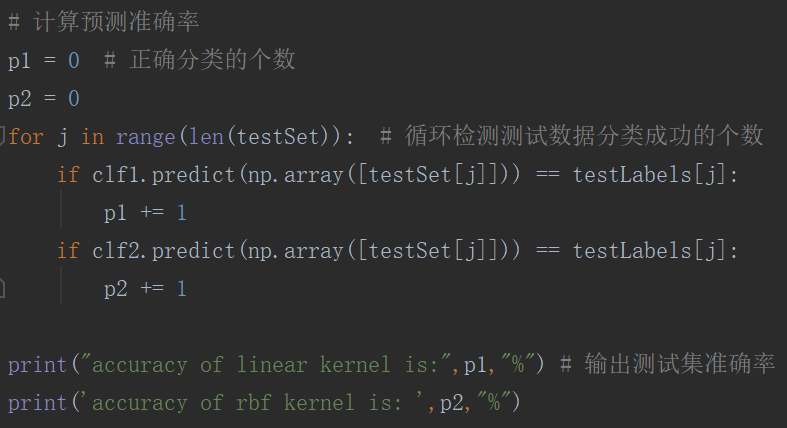
划分数据集：



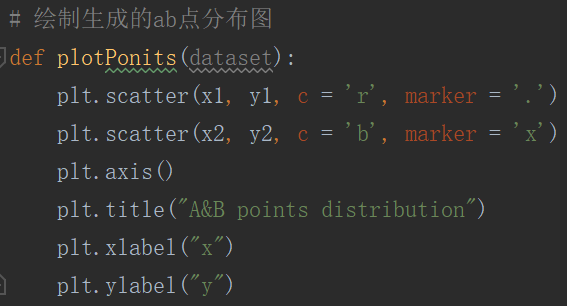
生成SVM、优化参数：



计算准确率并输出：



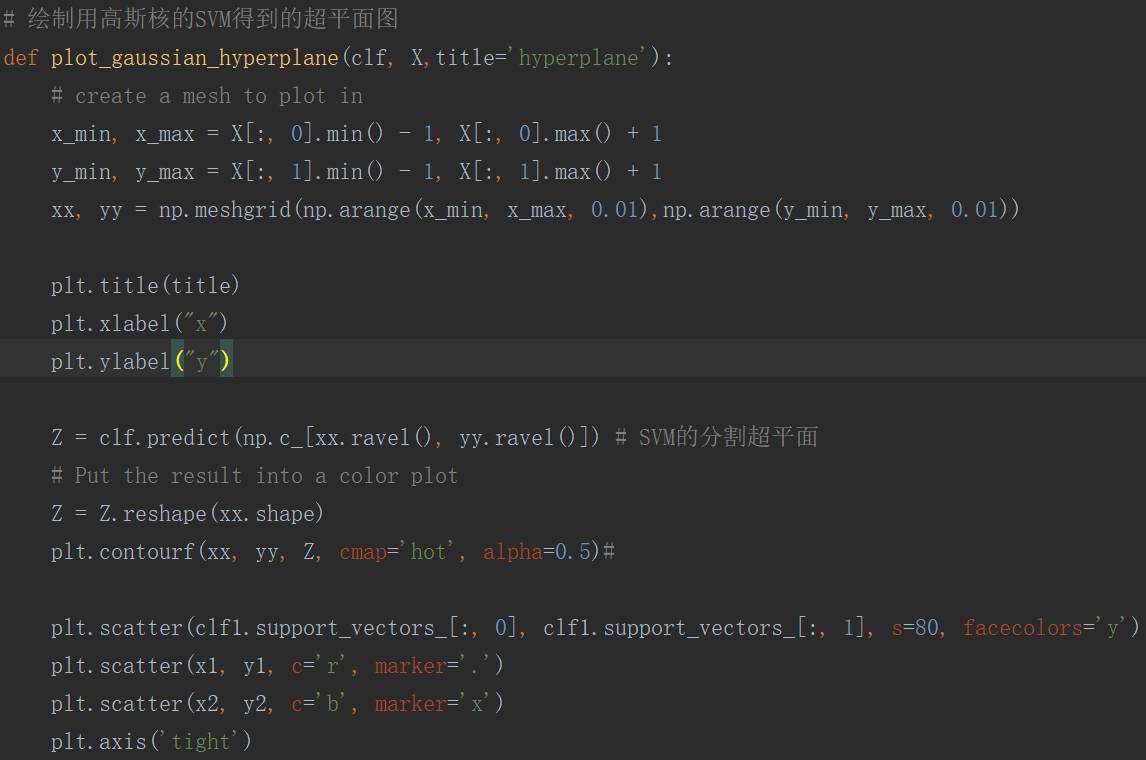
绘制生成的A、B点分布图：



绘制用linear核的SVM得到的超平面图：

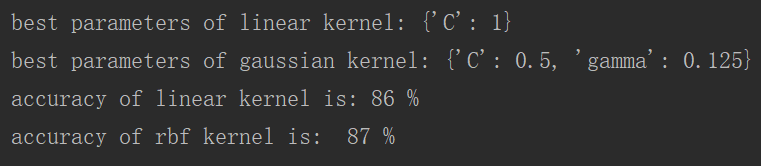


绘制用高斯核的SVM得到的超平面图：

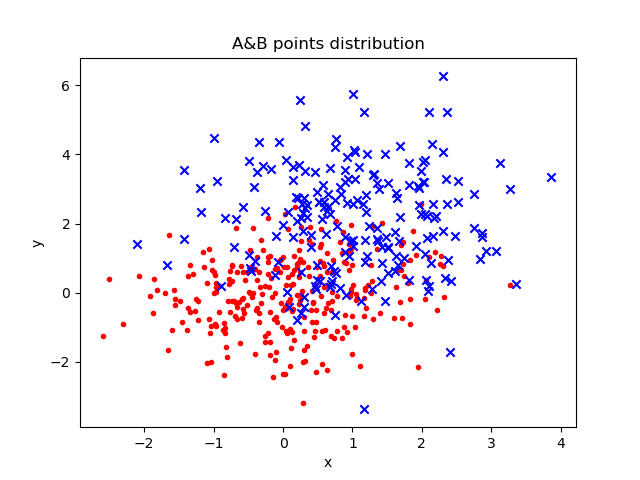


**实验结果：**

某一次测试的结果，输出了最佳的参数及对应的准确率：

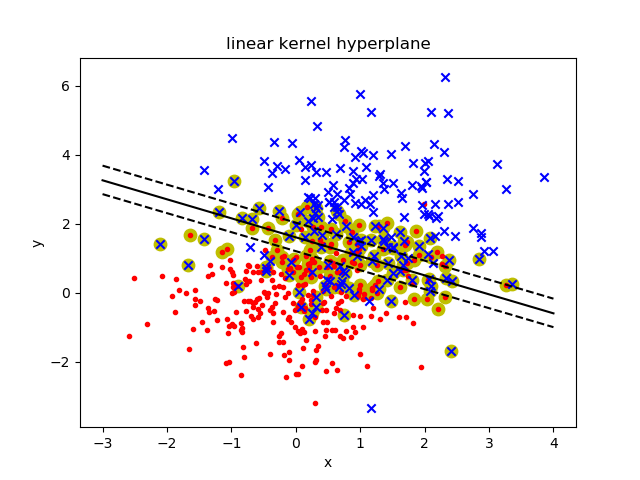


* AB点分布图：



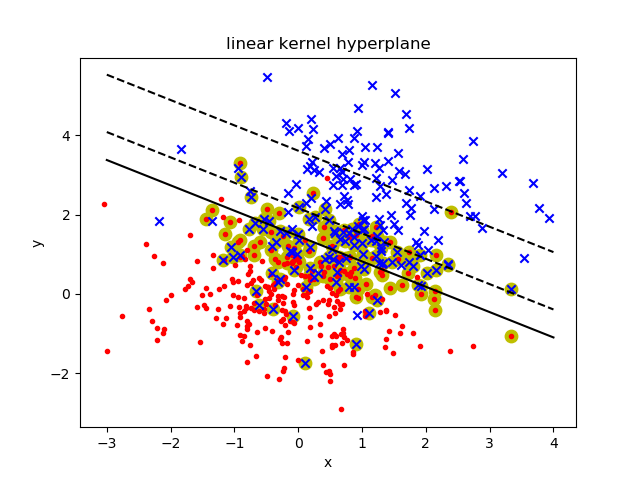
说明：A类点用红色点表示、B类点用蓝色叉表示。

* 用线性核的SVM得到的超平面图：

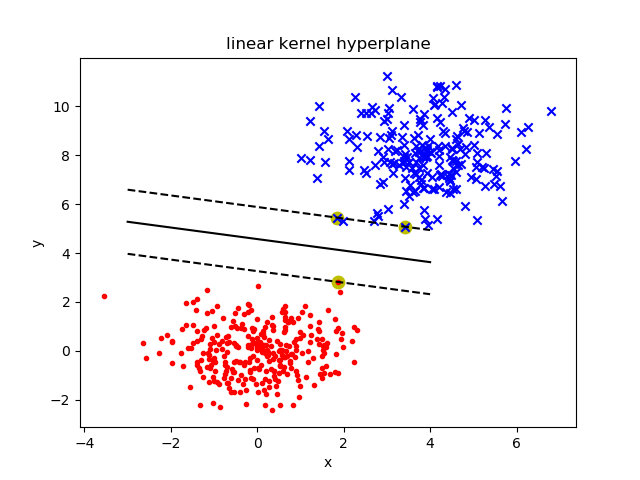


说明：A类点用红色点表示、B类点用蓝色叉表示。支持向量点外黄色圈圈包裹，黑色实线为超平面分割线，两条黑色虚线过支持向量的超平面分割线的平行线。

但在几次测试中，用线性核的SVM生成的超平面分割线并没有在两条过支持向量的虚线的中间，如下图所示：

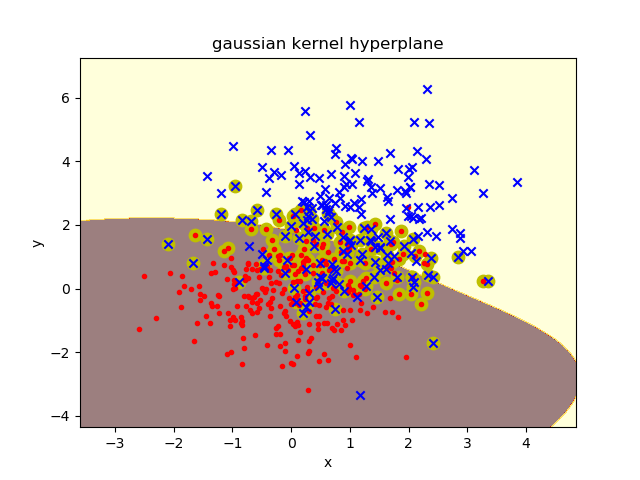


但若调整两类点的中心坐标，使两类点完全能线性可分，则产生的超平面分割线都能在两条过支持向量的虚线的中间。比如把B类点的中心调整为(3,6)，这样成了一个线性可分问题，此时用线性核的SVM出来的效果就非常好，分类的正确率也达到了百分之百。如下图所示：



我猜测是因为对于非线性问题，如果用线性核的SVM本身就会产生问题。

* 用高斯核的SVM得到的超平面图：



说明：A类点用红色点表示、B类点用蓝色叉表示。支持向量点外黄色圈圈包裹，下方的棕色区域为判定出为A类点的区域，上方的黄色区域为判定为B类点的区域。

**实验总结：**

这次的实验除了分类的方法与之前用的不同，最大的区别在于这次可以调用机器学习库中的方法了。这样的好处在于能利用python的强大之处，大大节省时间、提升效率，也更贴近现实中的编程过程，不好的地方也正是因为它是直接调用别人已经写好的，可能会让我们忽略了方法的本质。不过好在我们已经提前做了另外一个作业，对SVM公式的推导过程已经有了充分的了解。

有了之前这类问题解决过程的铺垫以及现成方法的调用，这次实验过程也显得相对轻松了一些。对于实验结果，我比较意外的是用高斯核的SVM测试出来的准确率不比用线性核的SVM测试出来的准确率要高多少，在好几次测试中甚至比后者要低。这明明是一个线性不可分割问题，照理来说高斯核会更好一些，我能想到的解释是可能参数还没有取到最优，或者这个点太多、两类点重复区域很大、分布实在过于杂乱。