**高斯朴素贝叶斯实验**

张靖祥 2017304010413 计算机172

# 1. 实验原理

已知：如下分类数据，不同品种的莺尾花有着不同的花瓣长度、宽度，花蕾的长度、宽度

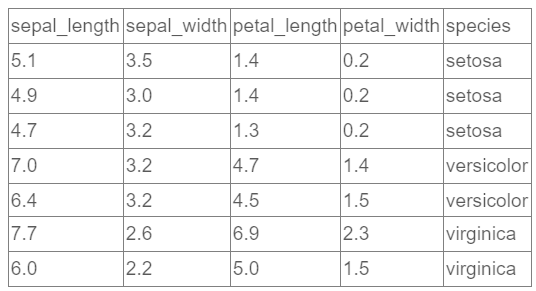
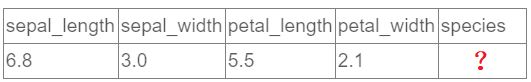
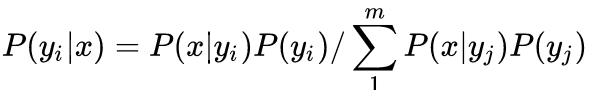


表1

现在有希望求得：

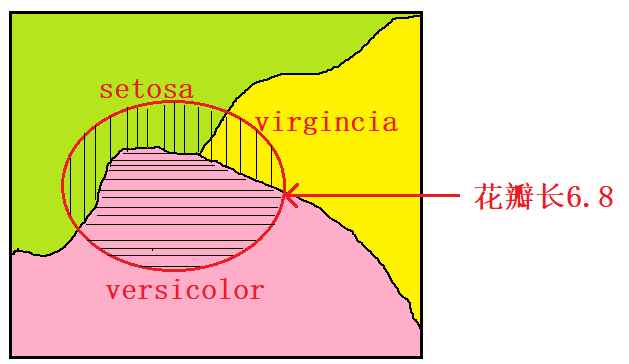


1. **线性代数中的贝叶斯的公式为如下：**



假设x是【花瓣长度6.8】，yi是【花的种类是vericolor】

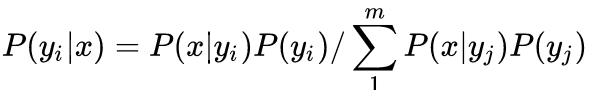
这个公式的意思是：在花瓣长度是6.8的**条件下**，花种类是versicolor的条件概率，实际上等于花瓣长6.8与花种类是vericolor的和事件的概率，除以花瓣长度是6.8的总概率。



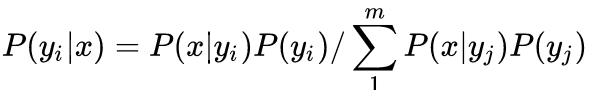
如上图中横线阴影占总阴影的比率。

1. **如果找到了最大的后验概率，也就完成了分类**

即：求max P(yi |x)，i为自变量。依次求出花瓣长度是6.8的情况下，花的种类是vericolor、setosa、virgincia的概率，找到最大的那个即是分类的概率。

现在，计算公式为：

Max 时候的i

等价于

Max 时候的i

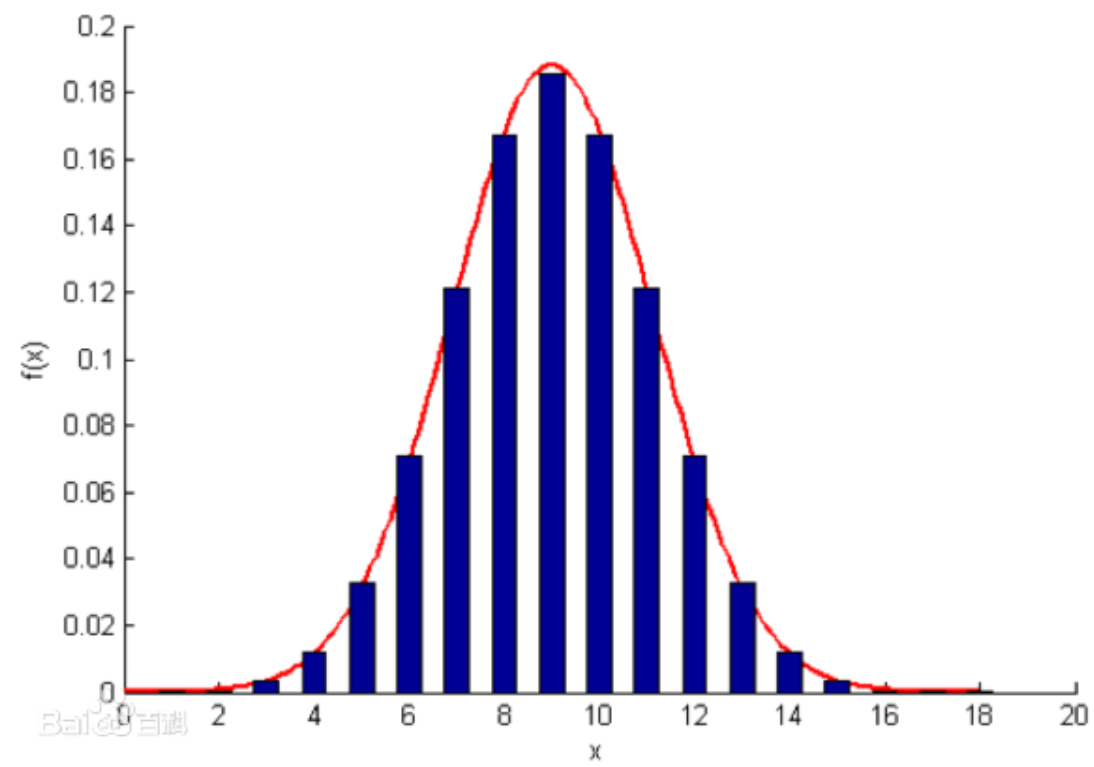
已知条件为表1

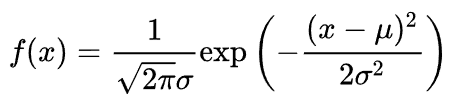
未知条件为：P(yi)，即：花种类是setosa的概率，virgincia的概率，vericolor的概率

P(x|yi)，即：在setosa种类的所有事件中，花瓣长度是6.8的概率。

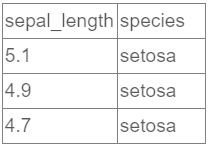
1. **高斯分布的引入**

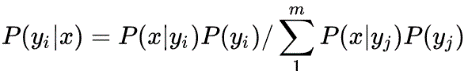
P(x|yi)，即：在setosa种类的所有事件中，花瓣长度是6.8的概率是无法求得的，因为花瓣长度的概率分布未知。现在假设花瓣长度的分布服从高斯分布，就可以将问题转化为求解高斯分布





首先，根据最大似然函数（不在展开）求出虽符合下表数据的高斯分布的参数μ和σ



根据求得的高斯分布的函数，将6.8带入，就可以求得花瓣长度是6.8的概率，即P(x|yi)，乘上setosa花的概率即可得到 ，在叶子长度6.8的情况下，依次算出所有的花的概率。

1. **朴素的概念**

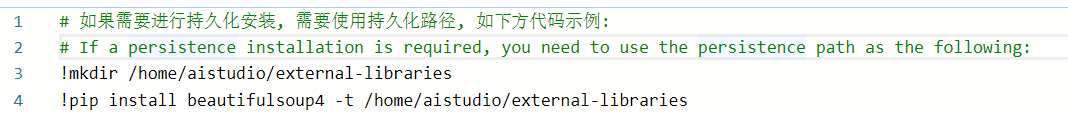
现在根据叶子长度这一项内容已经可以进行分类了，如果现在已知叶子的长度、宽度两项数据，那么首先要假定：**叶子的长度和宽度是独立的事件**，在这个基础上才能对概率进行乘积运算。因此就成为了朴素的方法，只能计算独立事件，相依事件是无法正确计算的。

之后只需要对每一个条件的概率进行乘积操作，取乘积的最大值即可。

# 2. 实验内容

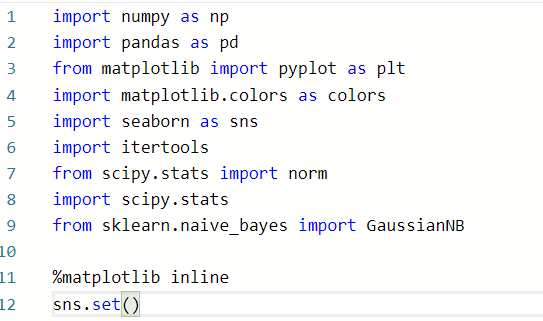
1. **实验代码的解析**

令人迷惑的代码之1：导入爬虫库



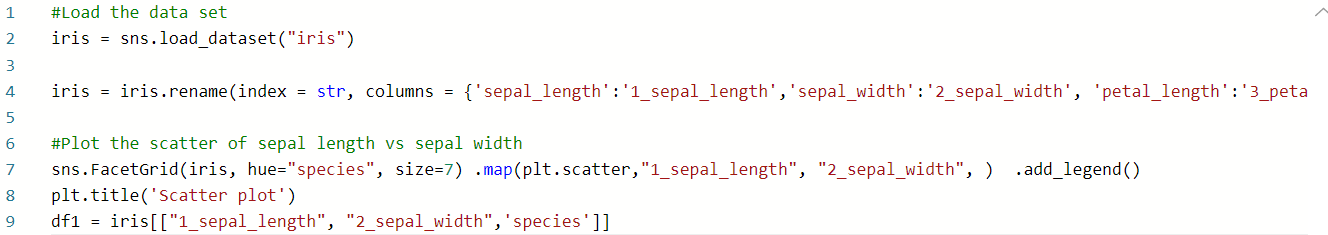
不知道用beautifulsoup是想要爬取什么数据

引入的类库：



之前一直听说过scipy是个科学计算库，一直没有亲自试用过

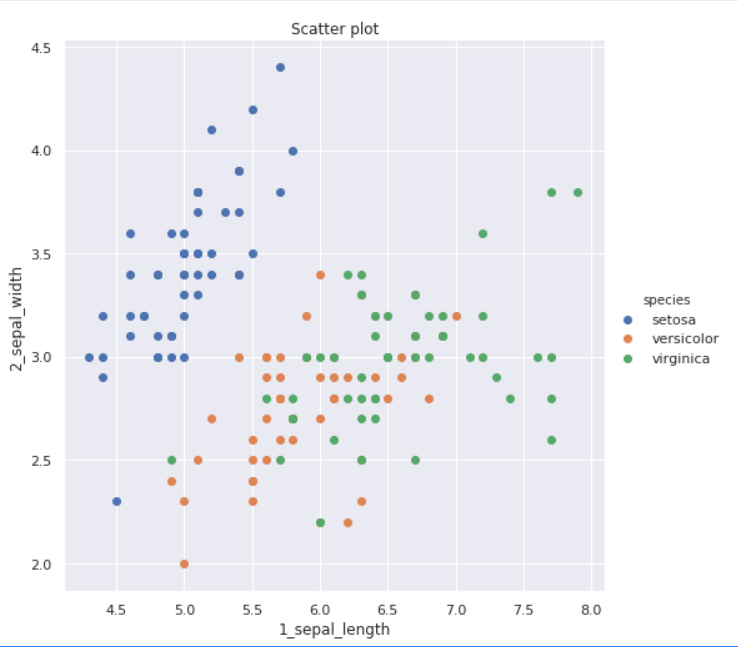
导入数据集：



巧妙的map使用方法，竟然把scatter这个绘制散点图的函数用这两个字符串去做映射，最后加上图示，这种代码的写作方法真是让人迷惑，强烈不建议这么写。

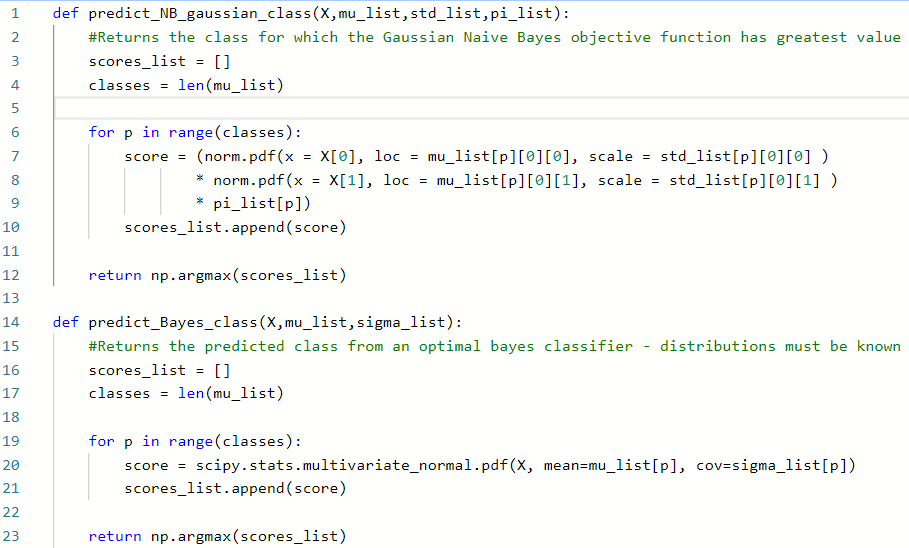
同时，这道题的用例使用的是叶子的长度和宽度，并没有使用另外两个参数，**因为参数超过两个没法画图。**

查看数据的分布：

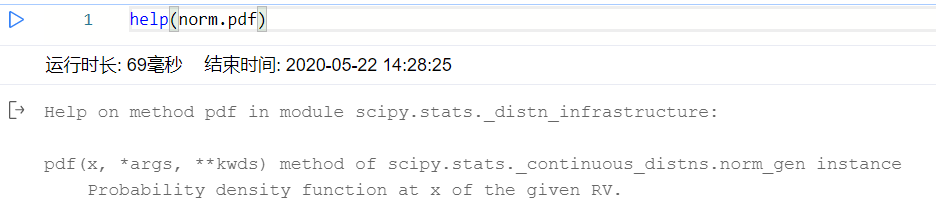
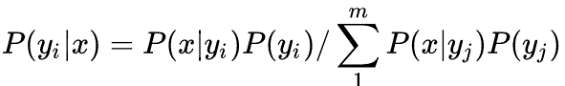


在1中已经分析了，高斯朴素贝叶斯是建立在独立变量上面的，但叶子的长度和宽度显然是不可能独立的，因此，绘制出的图像是可以看到叶子的长度和宽度是存在相依关系的。

函数计算：

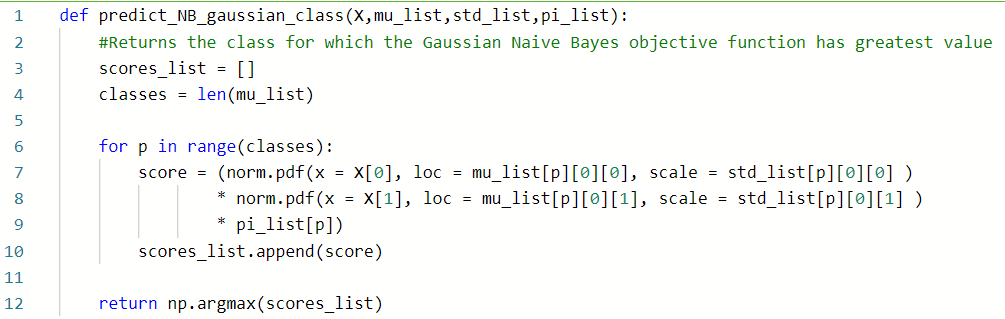


这里用到了norm.pdf函数，其详解如下

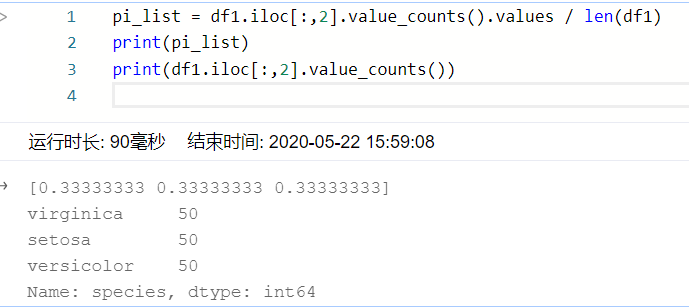
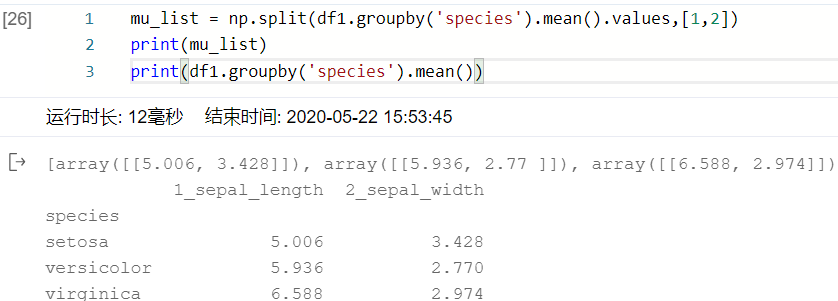


就是，输入一个正态分布的均值、标准差，求得概率密度，相当于 式子中的P(x|yi)

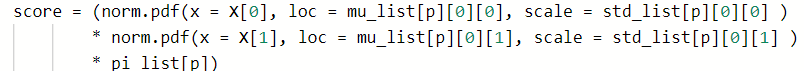
计算公式如下



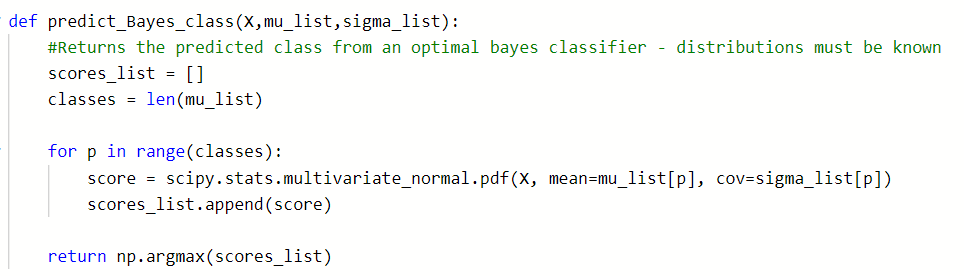
其中，输入的mu和std的list就是均值和标准差，pi则是P(yi)的列表.

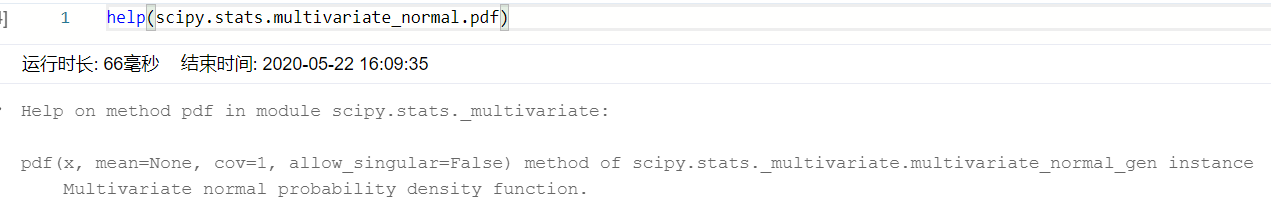
 

表达式中之所以是两个概率密度相乘，就是因为计算的是独立事件，作者将花的长度和宽度当成了两个独立时间，得到的概率密度相乘就是和事件的概率，当然，这种写法只适用于两个变量。



令人迷惑的函数2：计算多变量高斯分布的概率密度



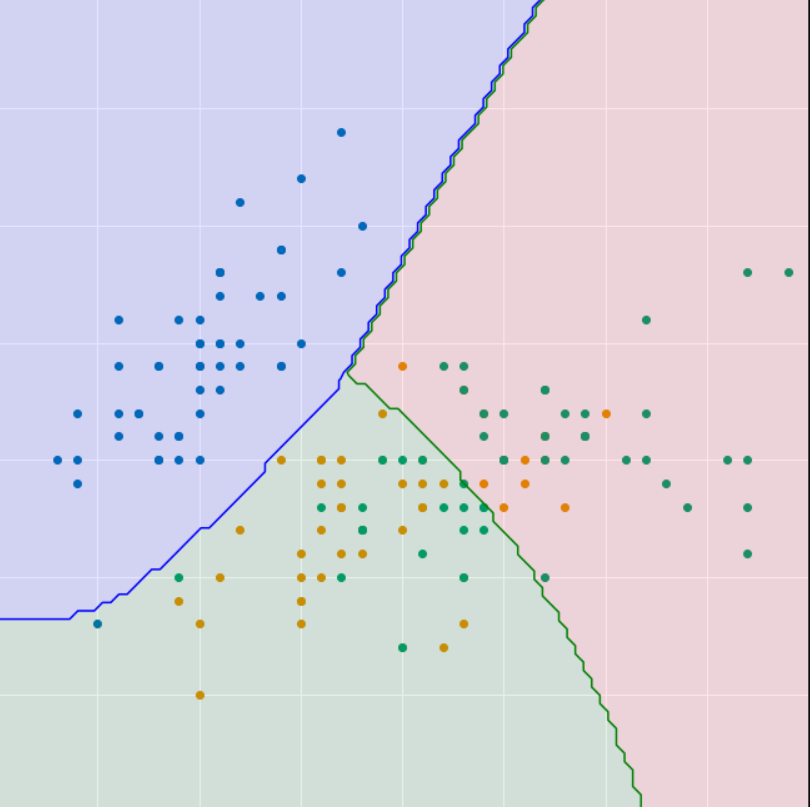


该函数的作用是，传入参数：希望计算密度的点x，平均值矩阵、协方差阵，计算得到概率密度

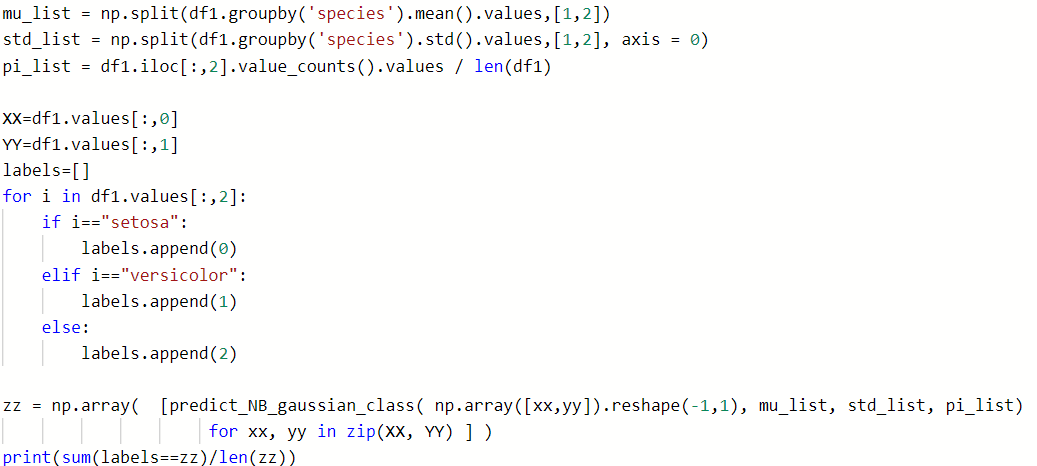
这个函数可以解决高斯朴素贝叶斯要求的独立分布的问题，但是很遗憾，拥有这么好的工具我却没有时间进行算法改良。

后面的代码没有值得分析的价值，基本都是绘图代码，关键的算法已经到此结束了。

1. **叶片长度和宽度进行分类的结果**



准确率验证代码：



准确率是：



# 3. 改进的可能与改进的方向

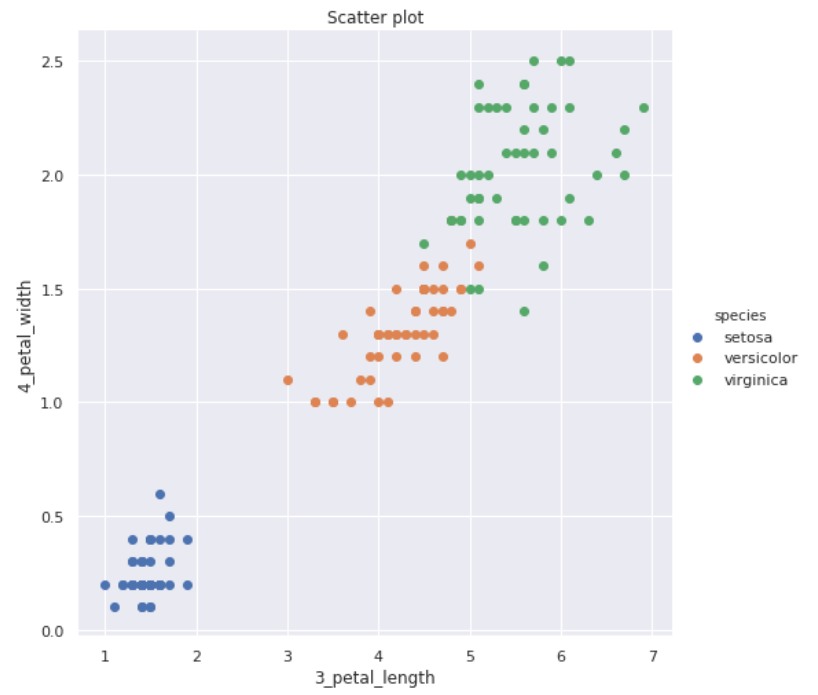
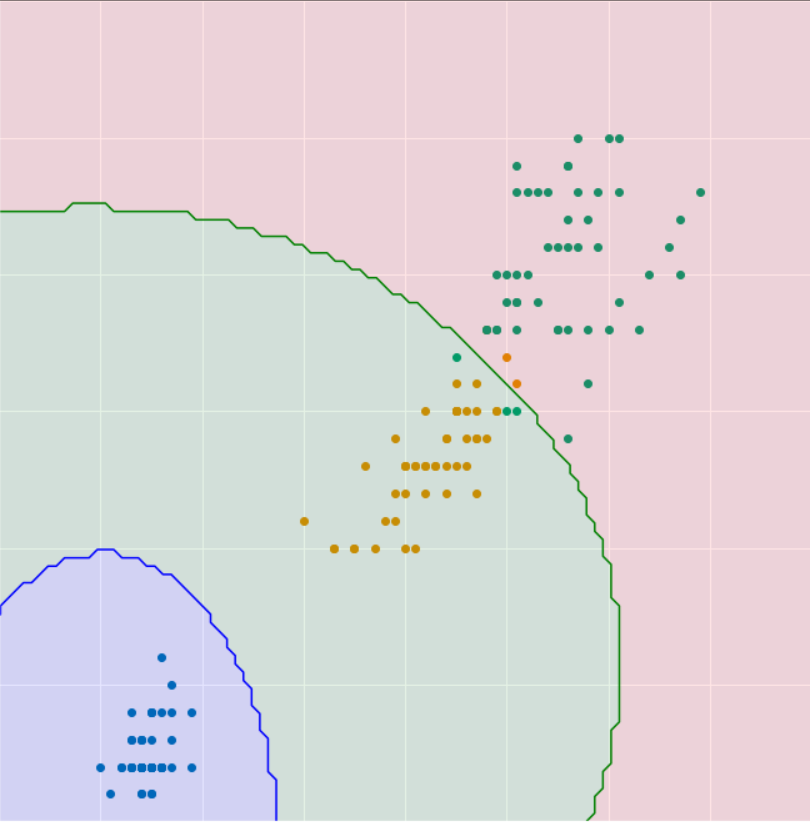
该实验可以改进的地方为：

1. 采用其他的变量进行准确率的测试，难度低
2. 采用其他多个变量，而不是双变量，但是不能画图，难度中
3. 采用非独立分布的多元高斯分布的概率密度进行计算，难度高

下面我从第1个和第2个的角度进行测试

## 3.1 采用其他变量进行准确率测试

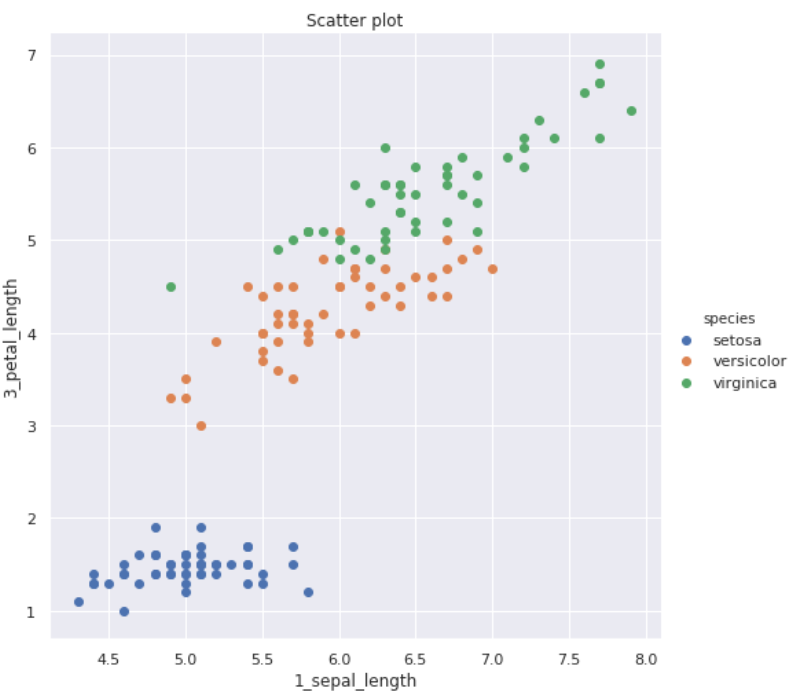
3.1.1 花瓣的长度和宽度（不是原始实验中的叶片的长度和宽度）

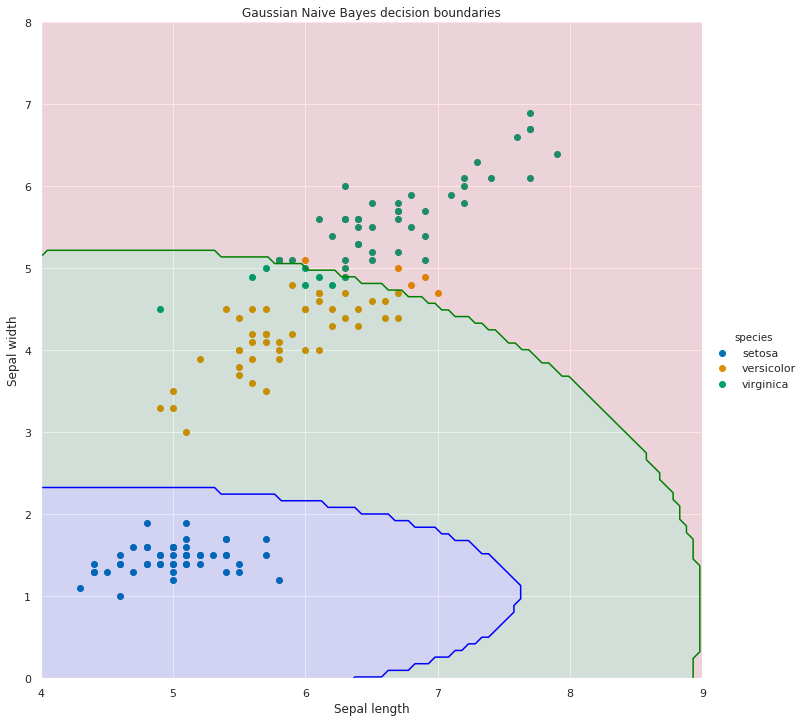
 

分类准确率为：



3.1.2 叶子的长度和花瓣的长度（不是原始实验中的叶片的长度和宽度）





准确率为：



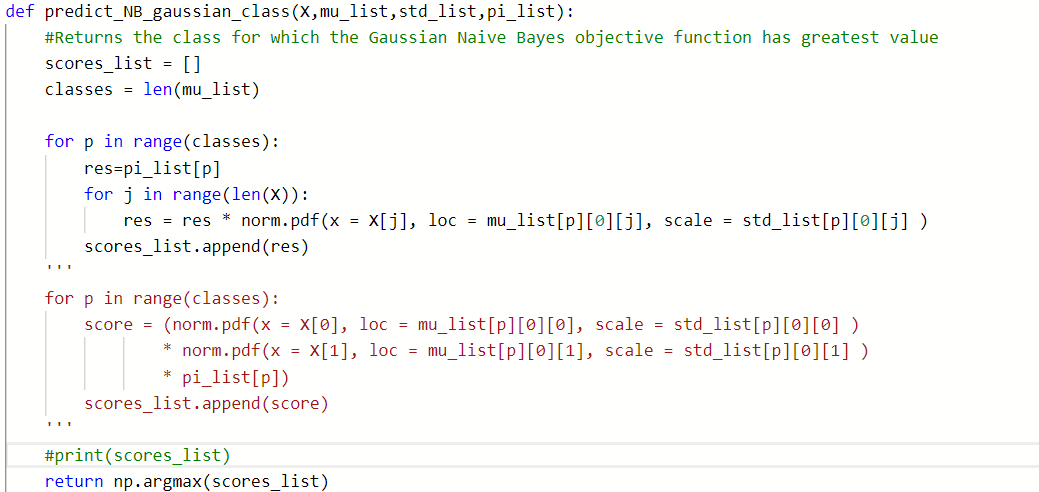
## 3.2 采用多变量

采用4个变量同时作为分类的依据

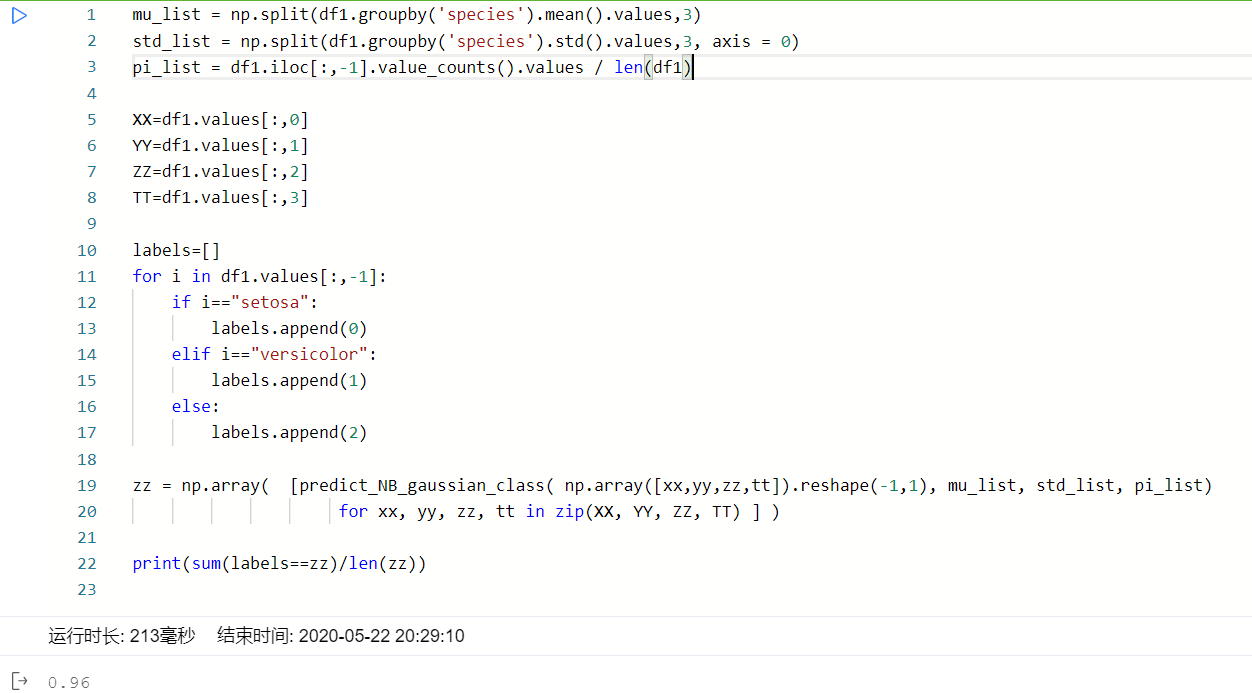


由于变量个数大于2，因此无法绘制图像

重新定义计算函数



识别函数如下



最终结果，准确率百分之96