目录

[感知机与最近中心分类器 2](#_Toc484967018)

[感知机 3](#_Toc484967019)

[最近中心分类器 5](#_Toc484967020)

[两个分类器比较 7](#_Toc484967021)

[类别不平衡问题解决 7](#_Toc484967022)

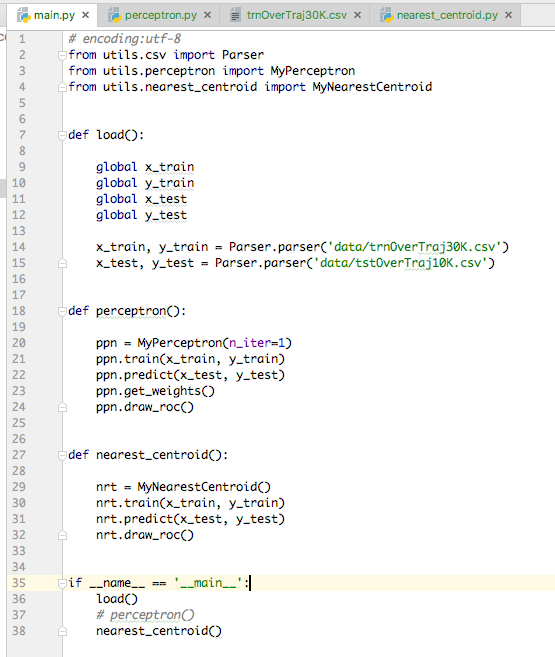
[欠采样 8](#_Toc484967023)

[调整类的权重 9](#_Toc484967024)

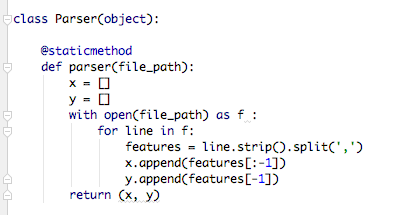
[调整决策的阈值 11](#_Toc484967025)

# 感知机与最近中心分类器

感知机与最近中心分类器都是分类算法的一种。调用部分的代码：



首先需要把数据加载，数据为csv格式的数据，解析比较简单：

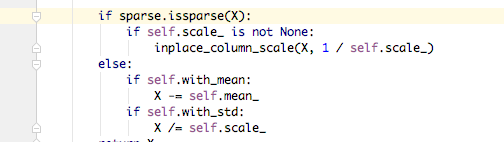


## 感知机

感知机模型根据训练集训练出分界面，一侧为正类，另一侧为负类。在本系统中，训练集有3w条，测试集有1w条。

首先对训练集的Input数据进行规范化（特征缩放与均值归一）：

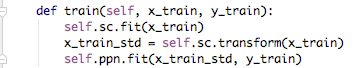




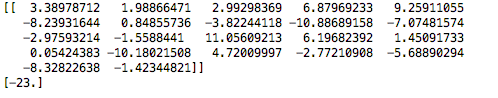
在调用时，设定的初值为学习速率：0.01，权重的初值均设置为1， 迭代次数设置为5：



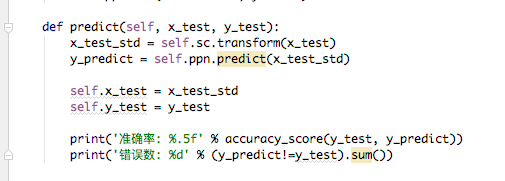
进行训练：



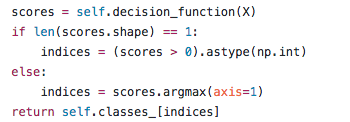
训练后可以得出权重w和常量b：



然后再把测试数据代入进行测试，



预测过程即是根据以上权重求出结果，与0进行对比：



分数是input与权重的点乘再加上常量：

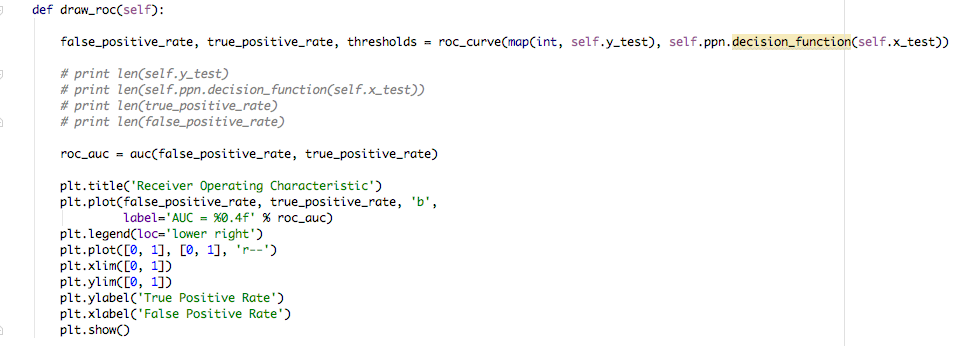


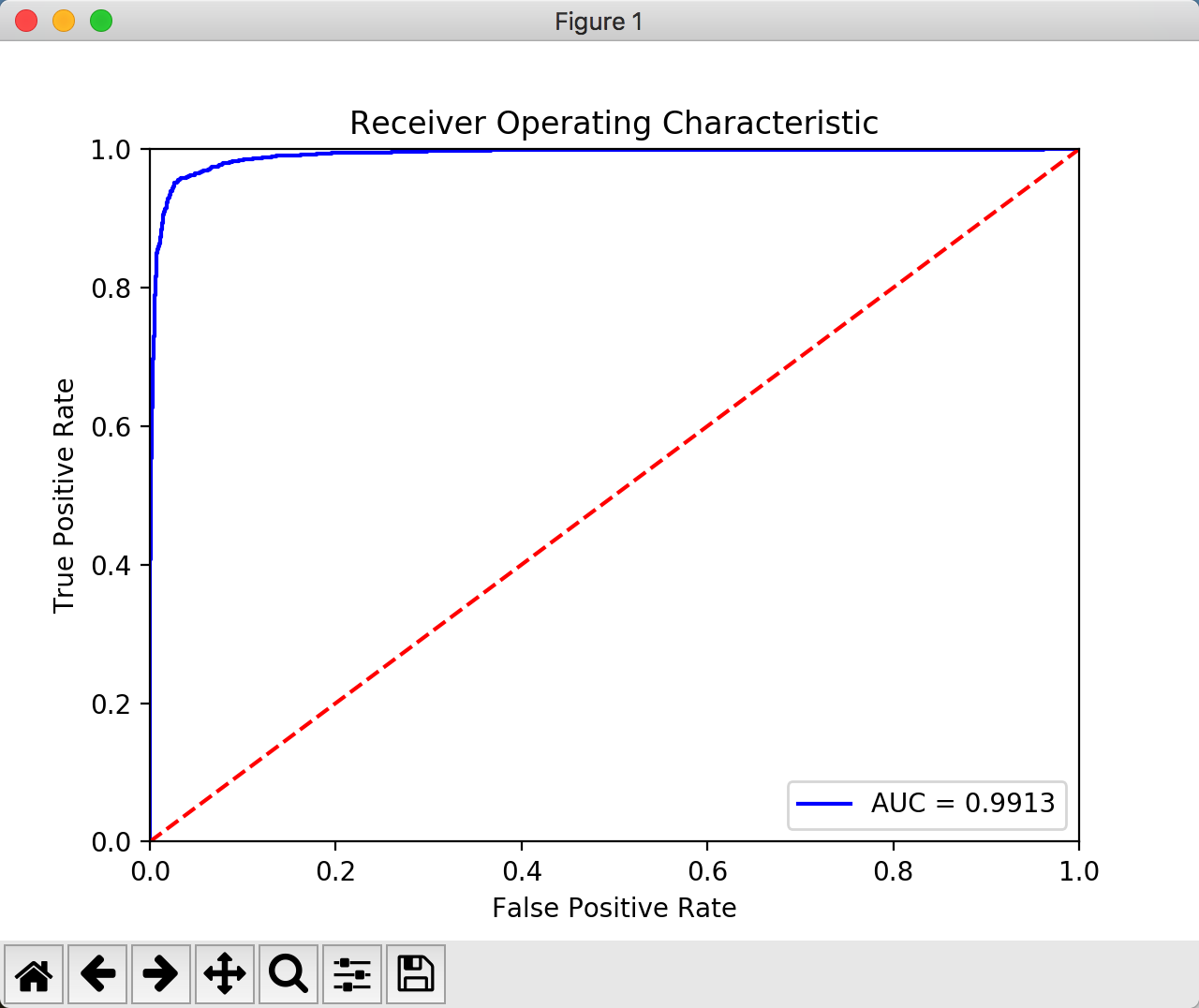
然后可以看到预测的基本结果：



绘制出ROC曲线：

ROC曲线是根据改变阈值绘制出的：

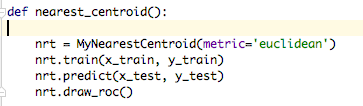




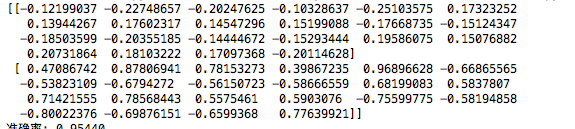
## 最近中心分类器

基本的步骤等同于感知机，只是模型本身和ROC 的评价本身不同。

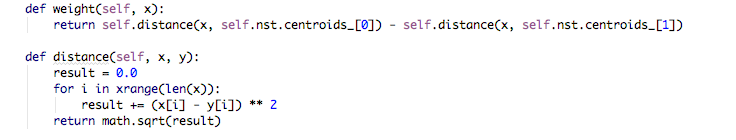
最近中心分类器的距离评测有多种标准，这里使用课件中提到的L2距离：即欧几里得距离。



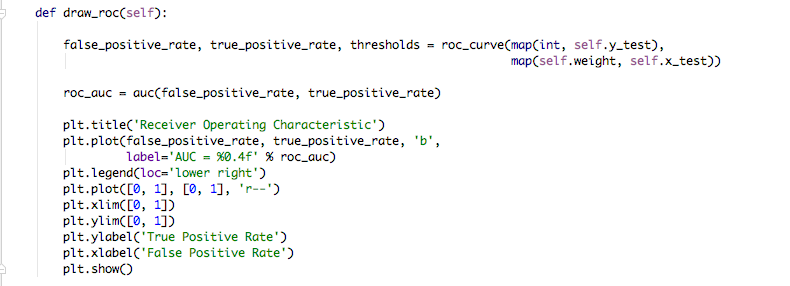
训练模型后，得到的模型即为两个22维坐标：



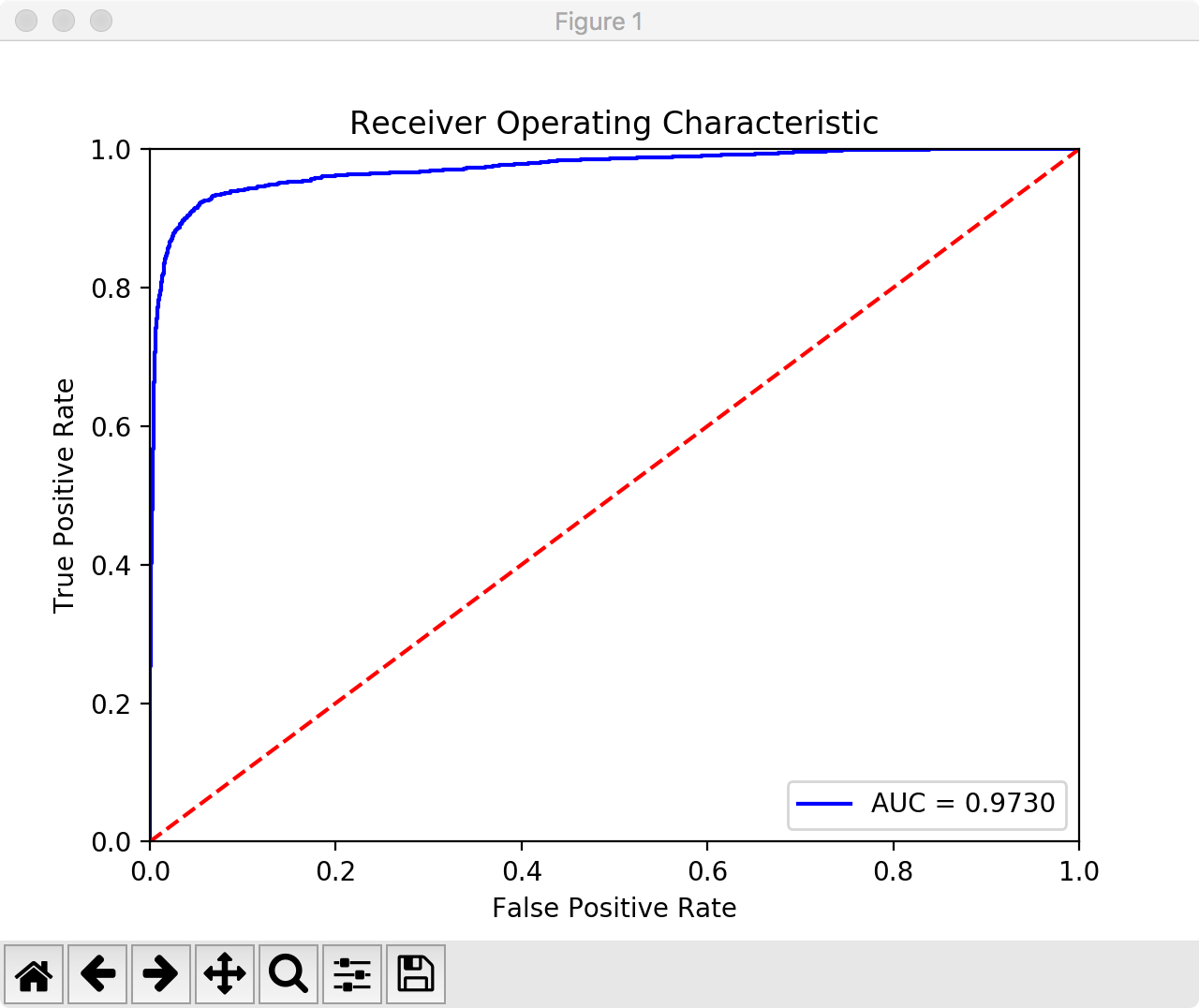
通过对两个距离作差，然后将0作为阈值，可以求得分类结果：



绘制出ROC曲线：



结果：



## 两个分类器比较

根据两个ROC图，明显可以看出感知机分类器效果更好，不管是TPR与FPR的比例，还是AUC面积值，都可以认为感知机在该数据该问题下表现地更好。

# 类别不平衡问题解决

在本数据集中，正类与负类的比例是1：4，这就导致即使我们把所有的数据都判别为负类，对于测试集而言，就已经有个80%的正确率。这显然与训练模型的初衷不符合。所以我们要首先保证正类尽可能少的错分。

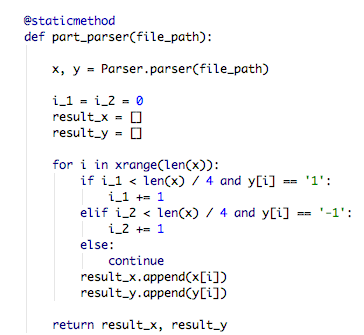
解决类别不平衡有一些策略：

1. 对正类过采样，就是创造更多的正类
2. 对负类欠采样，删去部分负类
3. 调整决策的阈值
4. 调整类的权重

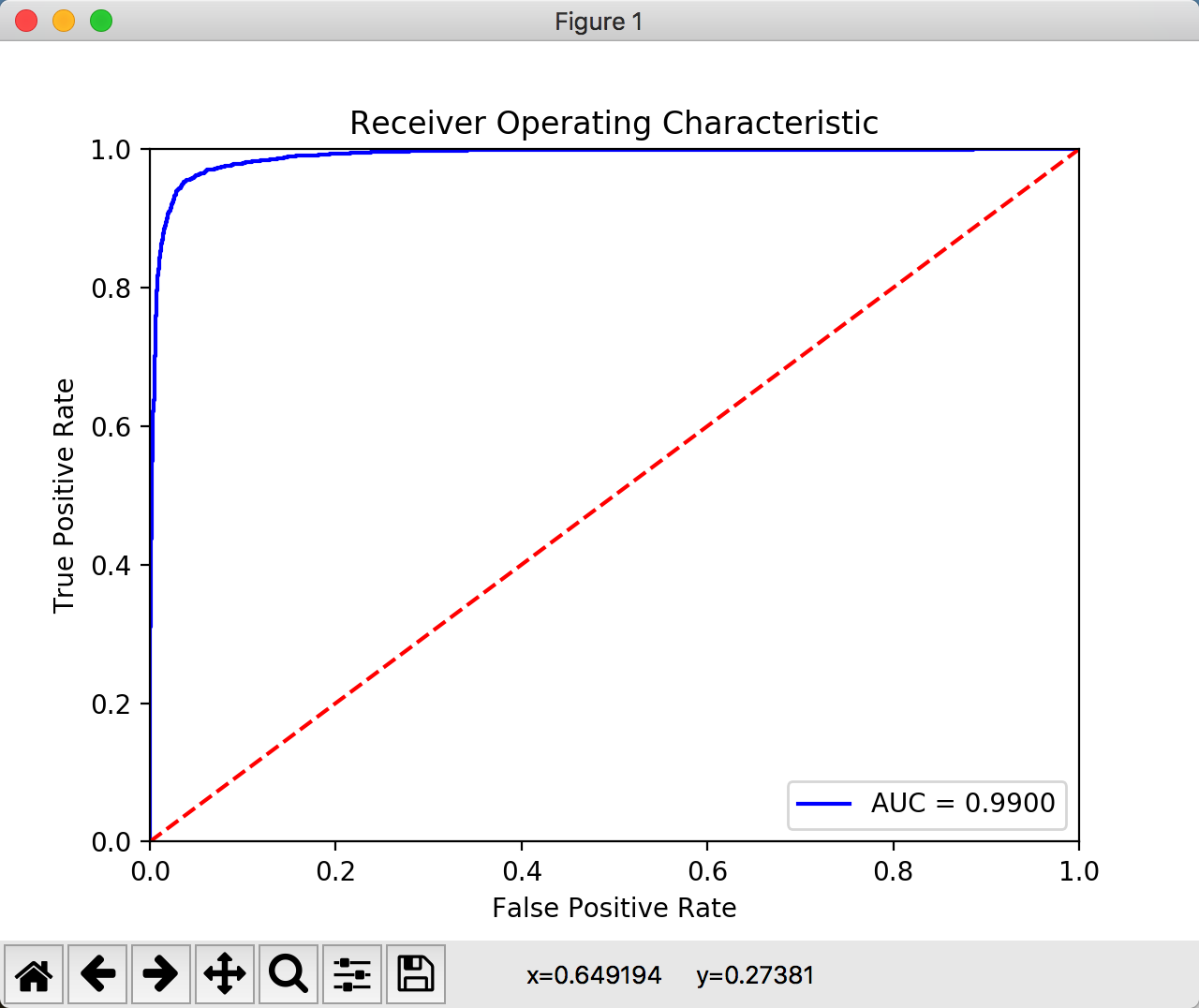
这里选用其中一些策略进行测试。

## 欠采样

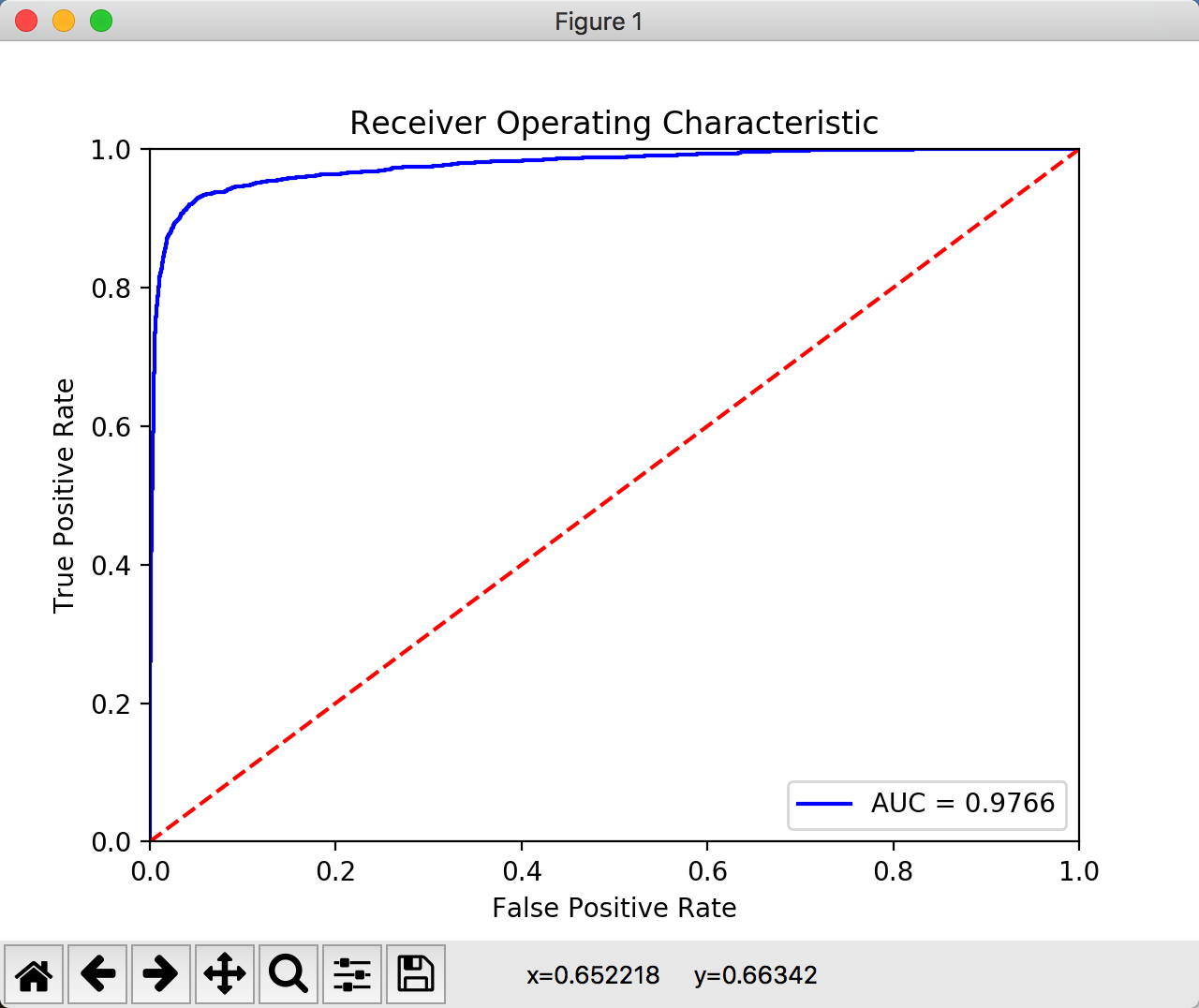
由于正：负=1：4，所以我们删除负类中多余的“3”部分：



经过测试，我们发现对于感知机而言，差距不是特别明显，可能是因为感知机本身效果已经很好：



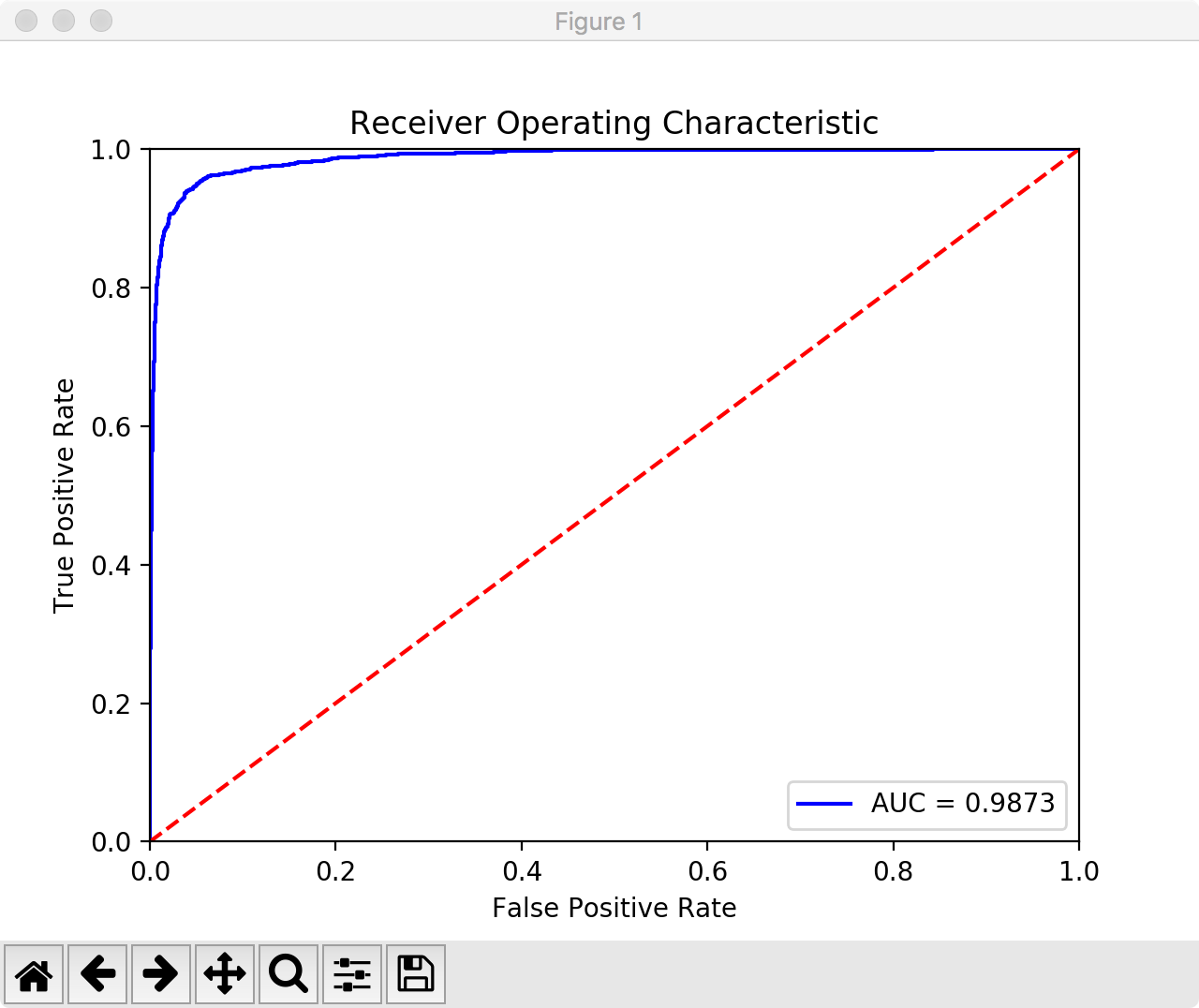
然后我们对最近中心分类器进行测试：



从测试结果中看，对于最近中心分类器的效果比较明显，其性能经过改善后已经基本贴近感知机的性能。

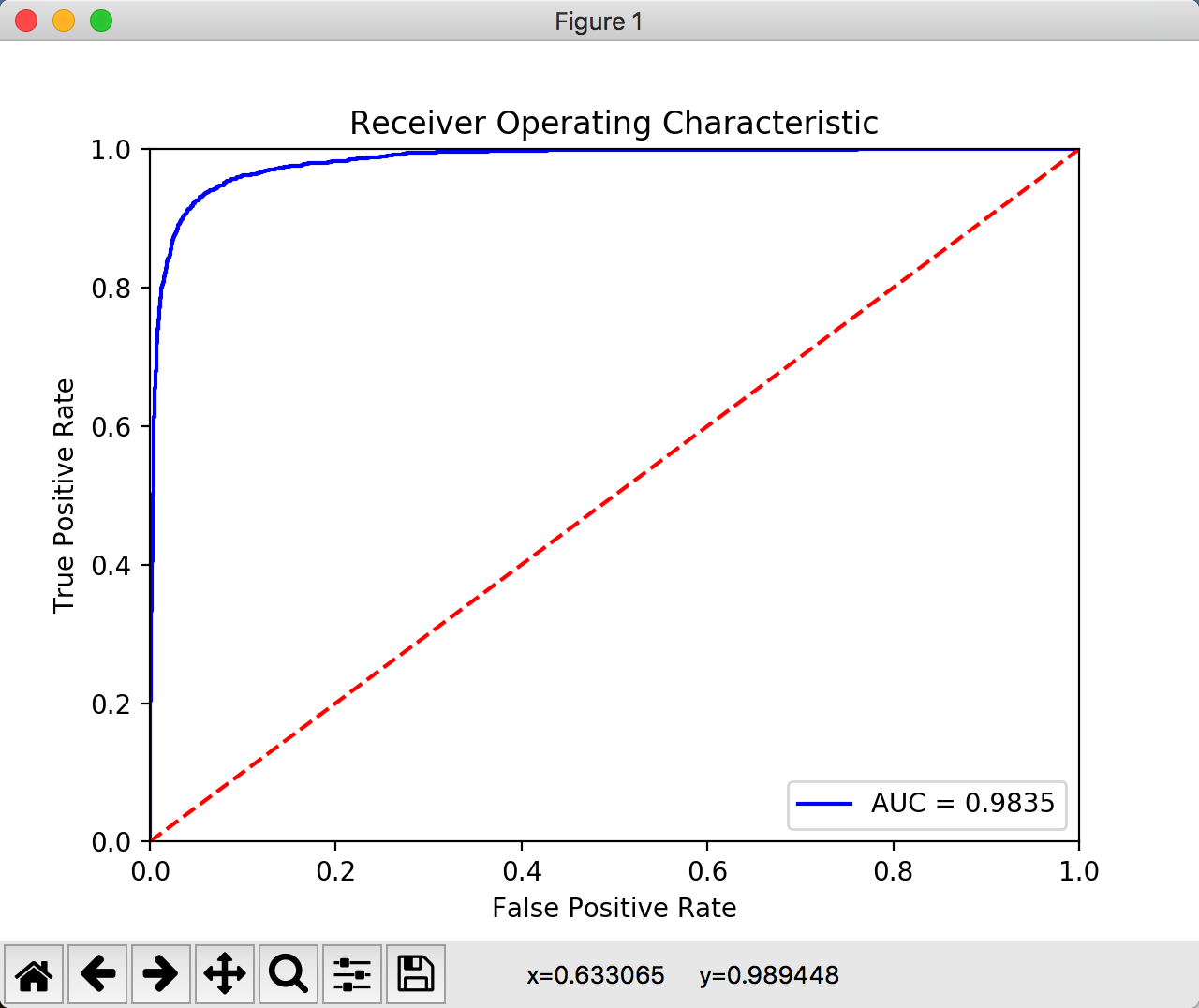
## 调整类的权重

在训练数据时，我们把两个类的权重设置为1：1，但是实际上我们更关心正类的正确率，所以我们可以调整权值，令其为4：1，让正类对训练权重的影响更大。

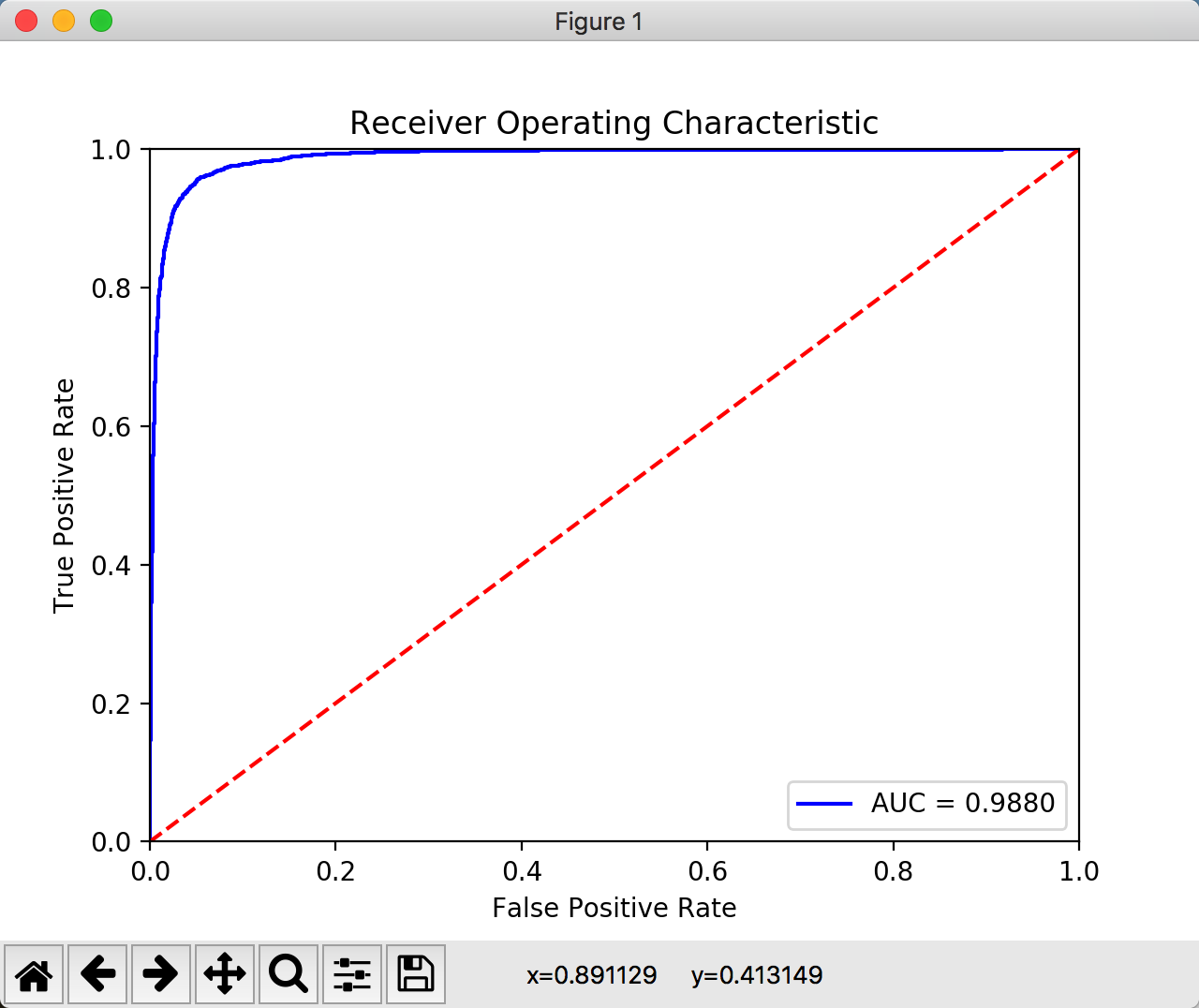


我们可以看出调整为4：1后，实际效果更差了。

然后我们尝试调整为10：1：



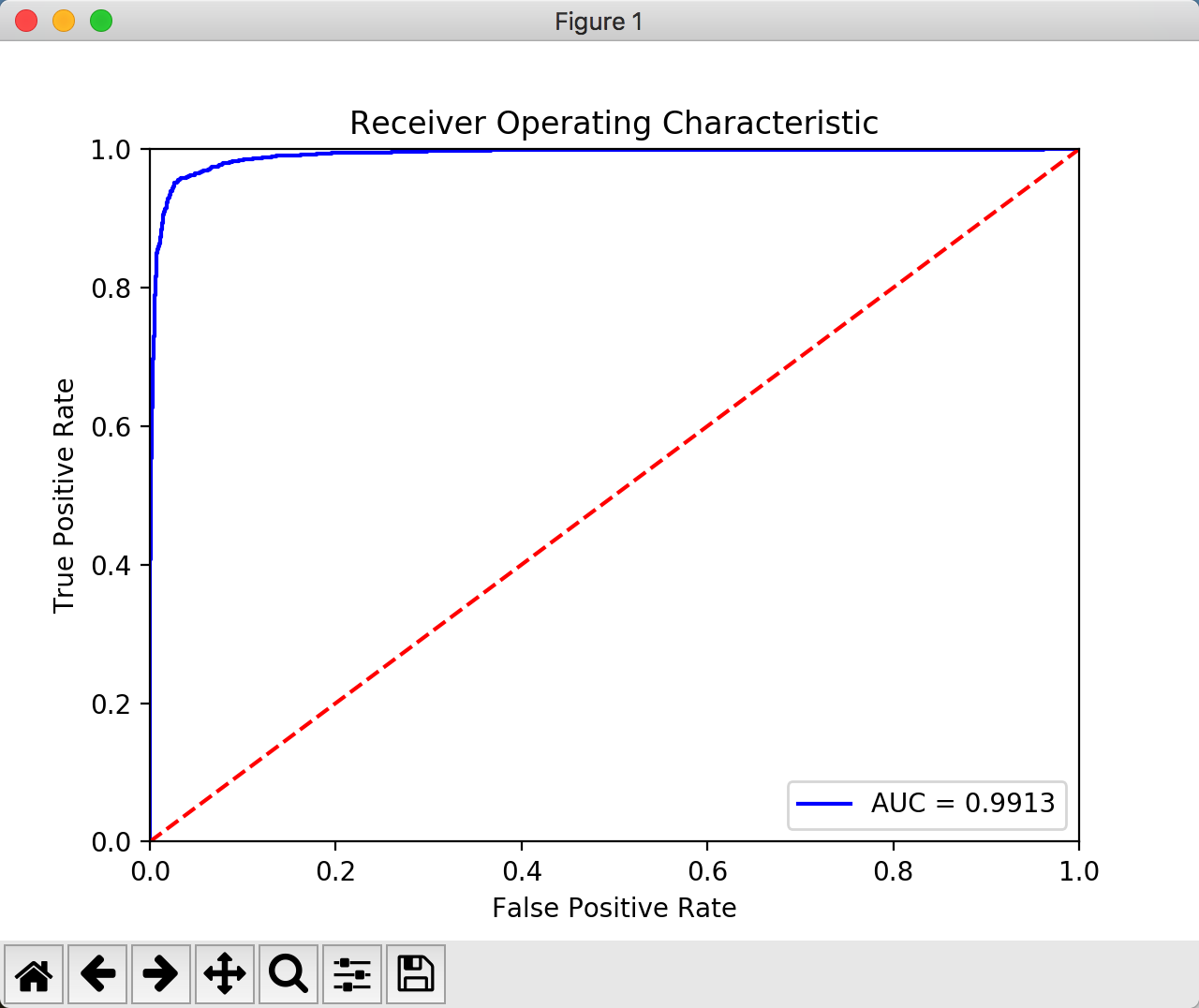
效果依然不好。再次尝试100：1:

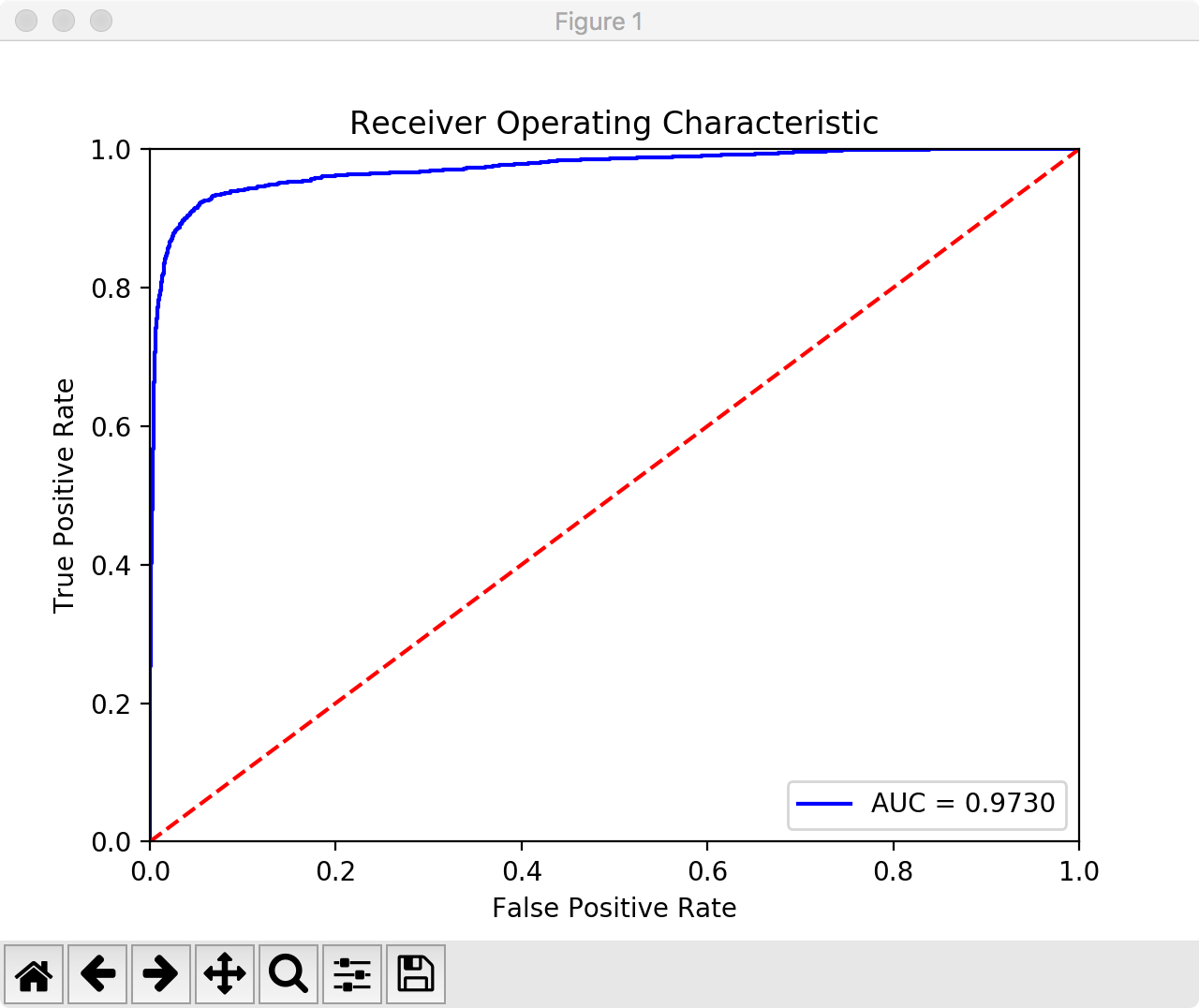


结果证明这个方法对于该数据集效果并不好。

## 调整决策的阈值

在ROC图的绘制中，我们可以看到调整阈值对结果的影响。前面的两个ROC图分别是





在修改阈值时，我们希望尽量减少FPR的值，这样就有更少的正类被分错，所以我们可以选择贴近y轴的点对应的阈值。