# 比赛报告

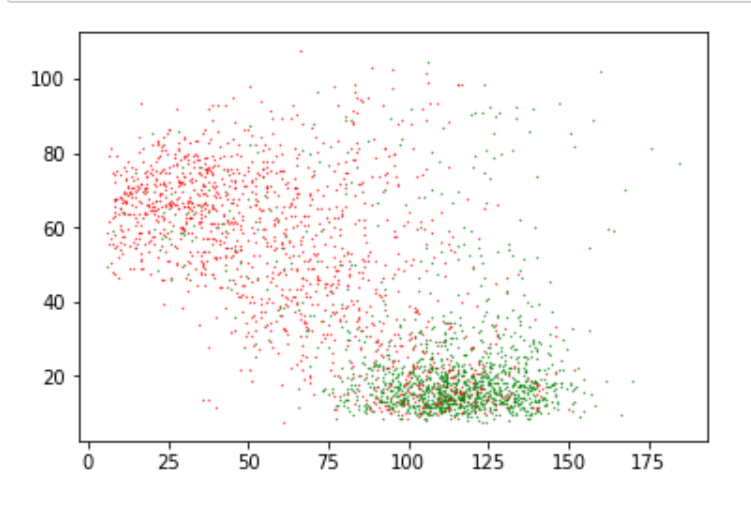
1、数据来源

训练集：HTRU\_2\_train.csv

预测集：HTRU\_2\_test.csv

2、分析数据

数据样本分布图



从图像中有一部分的点分布的比较散乱，但大体可以分成两部分，初步假设模型。

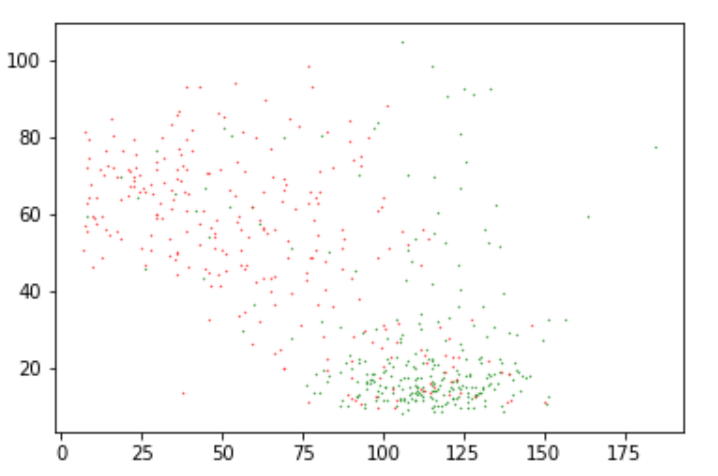
假设一：其他分布比较散乱的点为噪声点用逻辑回归模型将其成两部分。

假设二：发现有的红点在绿点的中心分布区域也有比较密集的分部，我想这些点不能当成噪声点来处理，所以使用knn模型或者贝叶斯模型来区分这些点。

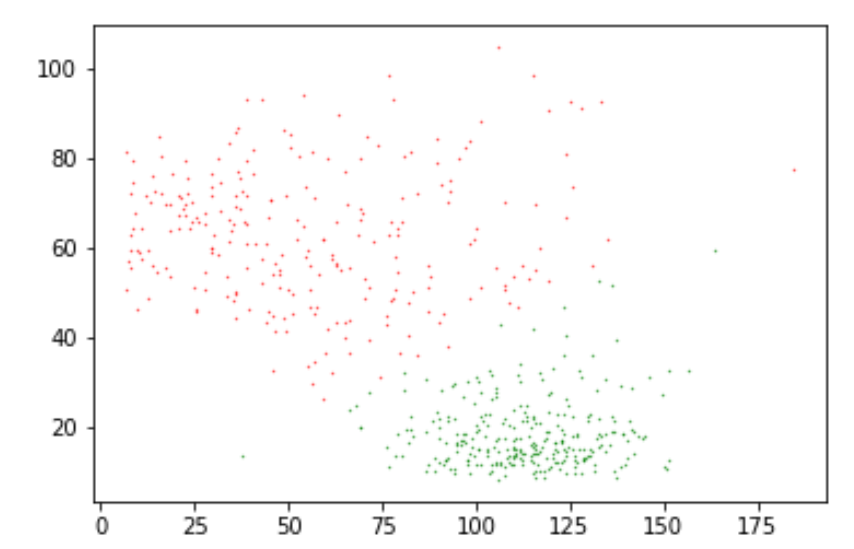
3、不同模型的分类结果图分析

首先将训练集分成：4/5的训练集和1/5的测试集

测试集原分布图



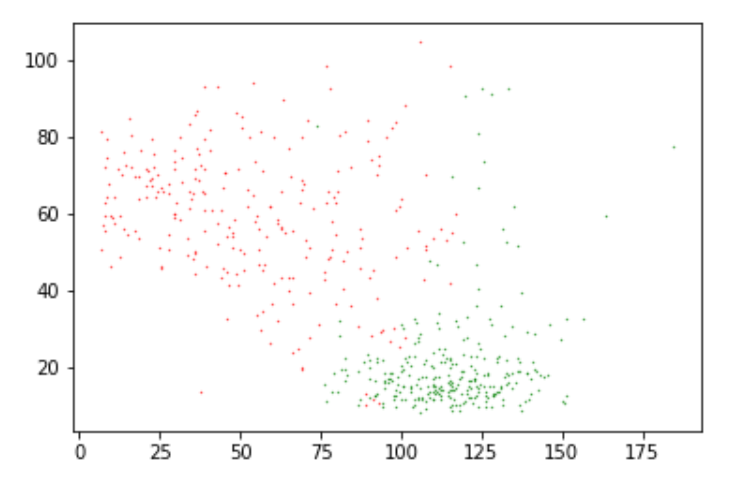
**逻辑回归模型**

（注:比较好的参数分出的结果）

分析：与原始数据进行比较发现可以正确区分出大量的点，但是它对一些位置比较特殊的点完全没得区分能力。

结论：简单的逻辑回归模型对特殊点没有很好的划分能力。

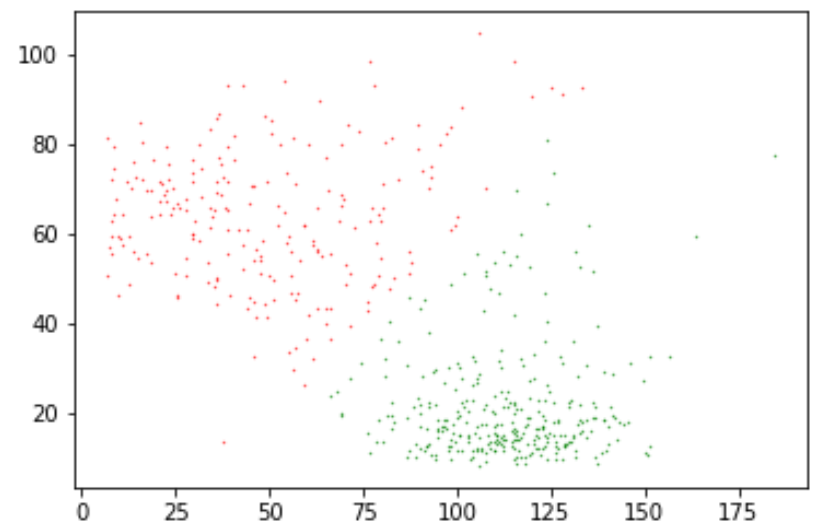
**KNN模型**

（注:比较好的参数分出的结果）

分析：此模型与逻辑回归模型相比较它能去正确的划分部分位置比较特殊的点，但是能力还是强。

结论：简单的KNN模型能对一部分特殊的红点进行划分。

**贝叶斯模型**



分析：结果和逻辑回归差不多，但是贝叶斯模型有个有趣的现象就是我将数据分成五份，进行五轮交叉验证发现贝叶斯模型的正确率非常稳定。

结论：简单的贝叶斯效果和逻辑回归模型差不多。

4、新思路及其方法

思路一：简单的让逻辑回归、KNN、贝叶斯分别去预测然后进行投票得出后的分类结果。

思路二：既然逻辑回归，简单的使用原样本效果不是特别好，那我就在数据集上对特征进行升维，然后再进行逻辑回归模型的训练。

思路三：使用逻辑回归和KNN训练出模型，然后用训练出的模型对训练集进行预测得到预测结果，再将预测结果和样本特征作为神经网络输入值，最会训练出一个模型来进行预测。

……

5、最终方法及其代码介绍

进过不断的尝试和比较最终发现knn效果还是不错的。knn的原理比较简单，来了一个带预测点，计算这个点到所有点的距离取其最近的几个点来对它进行投票。那类票多是那一类。

代码的实现在zmx.py文件中。对k值的选取在（数据的分析.ipynb）

（注：本文档中的所有图片，分析代码都在 数据的分析.ipynb）