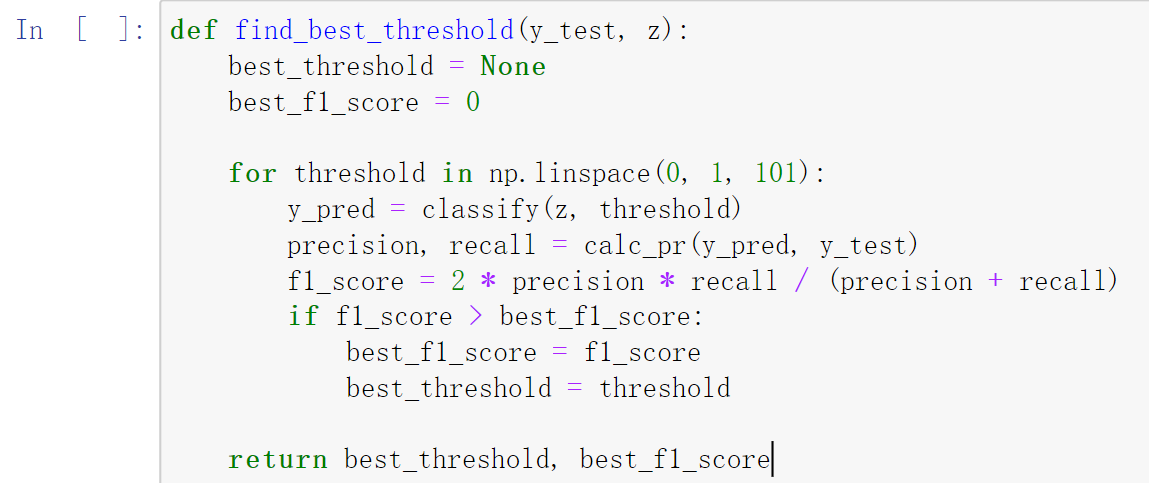
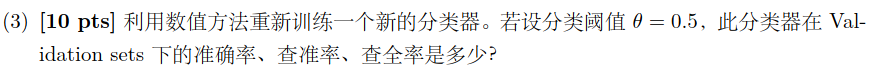


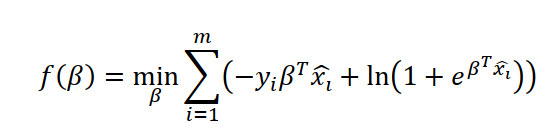
**通过F1-score可计算最优的阈值，在本实验中，经计算为0.5139247728112848-0.71920362729322429，在本实验中，我选择了0.7作为阈值进行实验，并把结果存在了/ML\_programming/58121128\_0.csv 文件中**

**以下是F1-score的python代码**

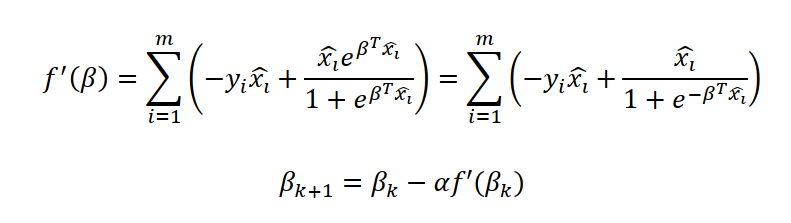




**我们的目标优化问题为**

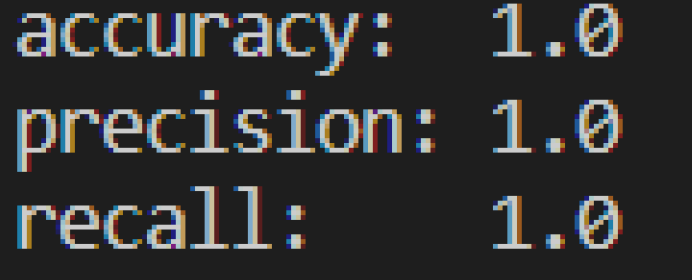


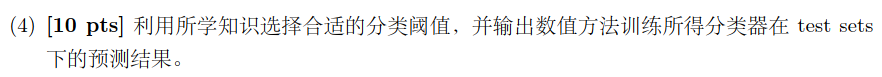
**我们在本实验中利用梯度下降法，具体算法如下**



**在本实验中，我们把学习率设置为0.05**

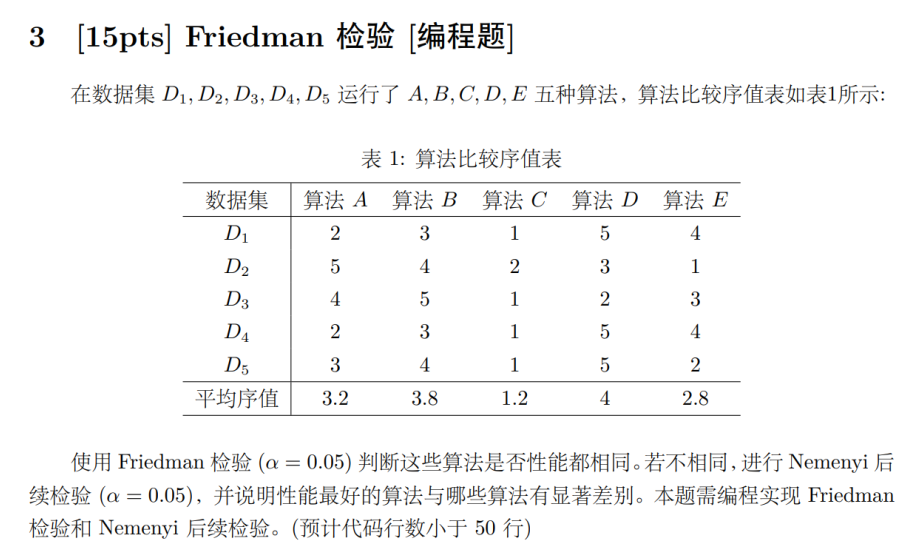
**最终结果如下**

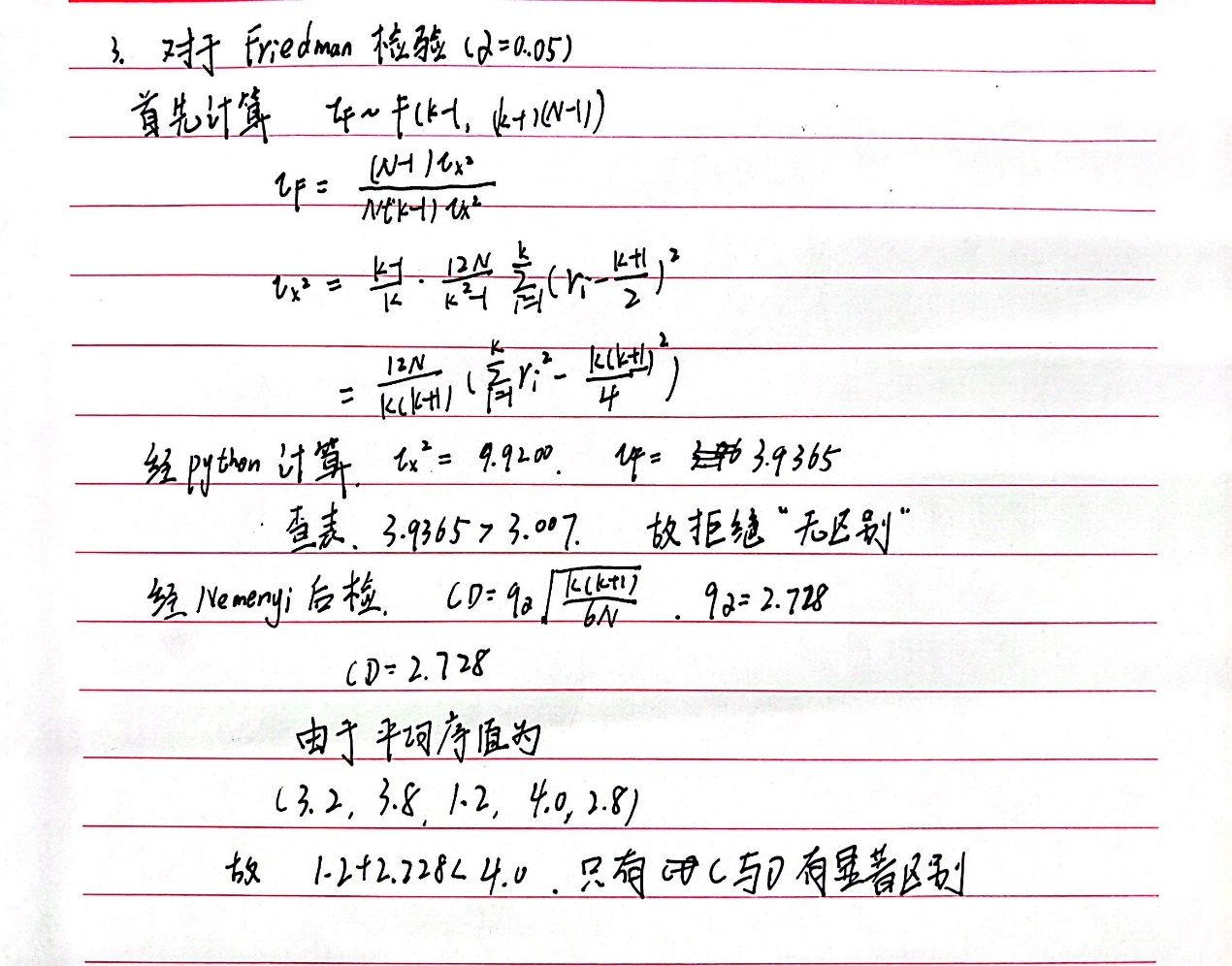




**同第二问**

  
**闭式解的意义是将标记预测为回归值，并近似y的值，而不是log的值。因此，在获取回归值后，可以使用任何激活函数来完成决策，例如tanh，sigmoid。由于在将mark用作回归值后，z的值更倾向于在[0,1]之间，而且在该问题中使用sigmoid函数（sigmoid函数不是增益运算），在z接近0时，梯度很大，也就是说，小误差造成的扰动会对sigmoid输出产生更大的影响，因此这种方法需要更精确的阈值。数值方法近似于速率值log，而z的范围不仅在[0,1]之间，而且当z远离0时，sigmoid函数的梯度很小，也就是说，小误差对sigmoid的输出影响很小，因此不需要非常精确的阈值。**





**代码**



