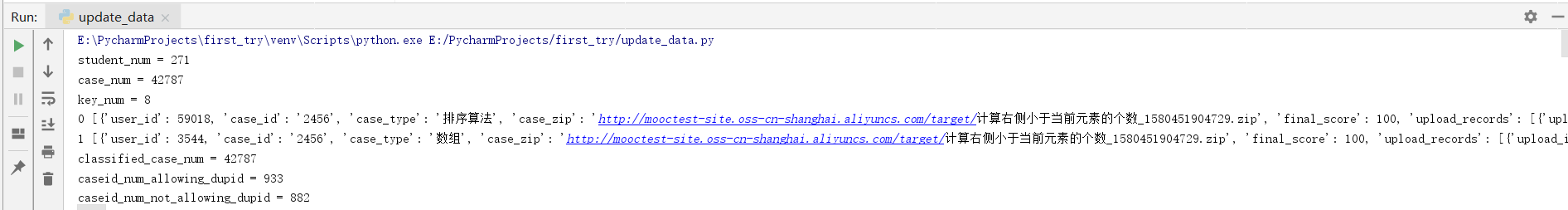
一、生成源数据集

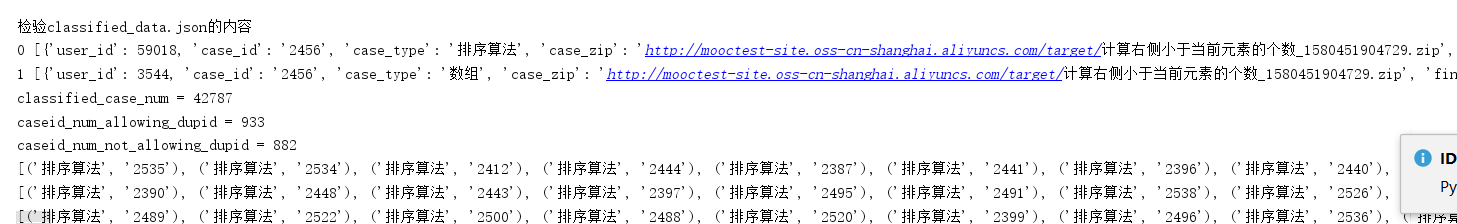
将test\_data.json按照八种题型分类时，发现不同题型的case\_id会重复，观察后发现不同题型的相同case\_id的题目完全一致，故而对学生的做题能力的度量也应当一致，于是归为一类。最终决定按case\_id分类test\_data.json，并在每个case\_id下的每个case中添加user\_id属性以备将来所需，生成实验第一阶段的源数据集。同时生成classified\_data.json作为第三阶段的源数据集。













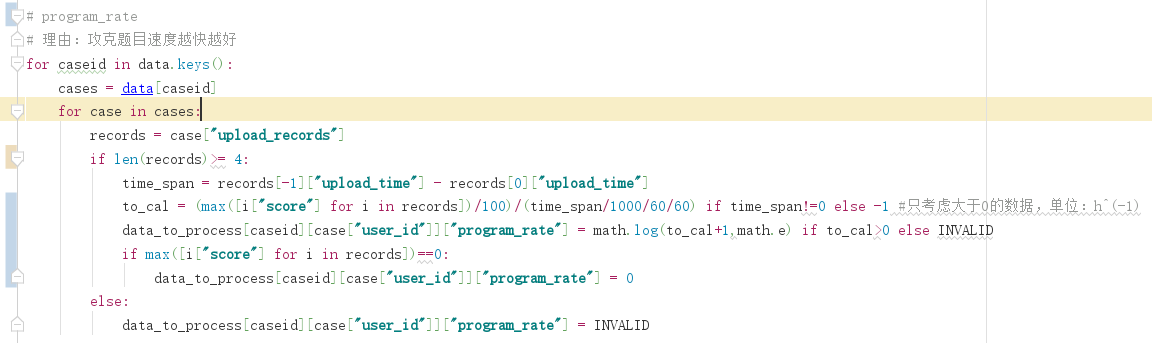
二、为第一阶段源数据集计算测量掌握值的指标

我们考虑测量编程者对题目的掌握值的指标，对于每个caseid的每个case计算该指标。由于过于简略的提交记录中所包含的做题信息是较为模糊的，对于某种指标，为了保证其合理性，只在一定提交次数以上的提交记录中计算，对于无法提取计算值的指标置为INVALID，便于在后续统计中筛去。

几种指标如下：

1. program\_rate：计算分数最大值与提交时间的极差的比值，将之代入ln(x+1)得出结果，若最高分为0则将结果置0

理由：我们认为分数最大值为编程者对题目的最终理解程度，而提交时间的总跨度可以衡量编程者的做题时间，包括think、program和debug的时间，代入ln(x+1)不影响单调性且结果最小值为0（在此指标计算中x最小值为0），而最高分为0的学生按最小值0处理。



1. debug\_rate：计算在一小时内的分数提升的几何平均数，将之代入ln(x+1)得出结果，若最高分为0则将结果置0

理由：限定只计算分数提升，以此期望所求的数值有更多可能是在原方法下攻克用例期间得到，而不是在试错或者转换方法期间得到，使得行为与debug更相符；限定计算几何平均数，使得每次分数提升都作为测量的一部分，与program\_rate的测量标准相区别；限定只计算一小时内的分数提升，使得所测量的行为与debug更相符。代入ln(x+1)不影响单调性且结果最小值为0（在此指标计算中x最小值为0），而最高分为0的学生按最小值0处理。

1. early\_success\_degree：计算到每次提交为止的最高分与满分的比值的均值，将之代入ln(x+1)得出结果，若最高分为0则将结果置0

理由：对于一段分数随次数提高的统计图，分数提高的越早，则可以体现编程者对题目的理解速度越快，故而用每次满分程度的均值来刻画此指标，使得分数提高越早时高分的权重越大、此指标的值越大；而对于分数下降的时候，我们认为这是编程者在尝试别的方法，其对题目的理解程度应保持与原来不变，所以仍采用之前分数的最大值计算。代入ln(x+1)不影响单调性且结果最小值为0（在此指标计算中x最小值为0），而最高分为0的学生按最小值0处理。



1. finish\_degree：计算第二次及以后的所有提交记录计算分数最大值与做题时间比值的均值，将之代入ln(x+1)得出结果，若最高分为0则将结果置0

理由：每次提交都是对做题者能力的一次评估，做平均后得出综合水平，可以衡量编程者包括思维能力与debug耐心等在内的综合编程能力。代入ln(x+1)不影响单调性且结果最小值为0（在此指标计算中x最小值为0），而最高分为0的学生按最小值0处理。



三、生成中间数据集

对于每项指标，以“含有有效指标的case数在15及以上”为条件筛选case\_id，得到每项指标的有效case\_id集合以及数量，对于每个case\_id中的指标数据判断是否满足正态分布，若满足的概率超过0.85则认为该项指标合理。



