**1.错误类型覆盖度**

数据集应包含多种常见的学生编程错误类型。如逻辑错误（如条件判断错误、循环终止条件错误）、运行时错误（如除以零、数组越界）边界条件错误（如未处理空列表、极值情况）、语法错误（尽管较少，但仍需适量包含）。

评价方法：统计数据集中不同错误类型的样本数量，确保各类错误样本的数量均衡，避免某类错误占比过大。

**2.错误修复难度**

数据集中错误的修复难度应多样化。如简单修复：单行代码修改（如变量名错误、运算符错误）、中等修复：多行代码调整（如循环结构错误、条件判断错误）、复杂修复：算法逻辑重构（如错误的排序算法、递归终止条件错误）。

评价方法：统计不同难度等级的样本比例，确保数据集覆盖多种修复难度。

**3.代码复杂度**

数据集中代码片段的复杂度应多样化。如代码行数：短代码（几行）和长代码（数十行）、控制流复杂度：简单的顺序执行、复杂的循环和条件嵌套、数据结构复杂度：列表、字典、嵌套结构等。

评价方法：使用代码复杂度度量工具评估代码片段的复杂度并统计数据集中不同复杂度等级的样本比例。

**4.数据集的标注质量**

数据集中错误的标注应准确和完整。如错误代码位置、错误类型、修复方案、错误影响范围（完全无法运行或者导致部分输出结果错误）等内容标注

评价方法：人工或自动化检查错误标注内容的准确性、完整性。