**Summary**

This report presents the results of static code analysis and test coverage analysis for the triangle\_classifier.py program. I used Pylint for static analysis and pytest-cov for test coverage to ensure code quality and reliability.

**1. GitHub Repository**

The analyzed code is stored in the following GitHub repository:

<https://github.com/MeiXu2910/helloworld>

**2. Static Code Analysis with Pylint**

I ran Pylint on triangle\_classifier.py, which initially returned a score of 7.00/10 due to the following issues:

* Trailing whitespace (line 3)
* Trailing newlines (line 15)
* Missing module docstring (line 1)

**Corrections Made:**

* Removed unnecessary trailing whitespace.
* Ensured consistent newline formatting.
* Added a module-level docstring.

After making these changes, we re-ran Pylint, and the final score improved to 10.00/10

**3. Test Coverage Analysis with pytest-cov**

We used pytest to run unit tests on triangle\_classifier.py and verified test coverage using pytest-cov.

Initial Coverage Results (Before Improvements)

|  |  |  |  |
| --- | --- | --- | --- |
| File | Statements | Missing | Coverage |
| triangle\_classifier.py | 10 | 9 | 10% |
| test\_triangle\_classifier.py | 12 | 5 | 58% |
| Total | 22 | 14 | 36% |

This indicated that a significant portion of the code was untested.

**Improvements Made:**

* Added test cases to cover all edge cases.
* Ensured all branches of the classify\_triangle function were tested.

Final Coverage Results (After Improvements)

|  |  |  |  |
| --- | --- | --- | --- |
| File | Statements | Missing | Coverage |
| triangle\_classifier.py | 10 | 0 | 100% |
| test\_triangle\_classifier.py | 12 | 0 | 100% |
| Total | 22 | 0 | 100% |

4. Screenshots of Analysis Results

Before Fixing Errors (Pylint Score: 7.00/10)

文本

描述已自动生成

After Fixing Errors (Pylint Score: 10.00/10)文本

描述已自动生成

Initial Coverage Report (36%)

表格

描述已自动生成

Final Coverage Report (100%)

文本

描述已自动生成表格

描述已自动生成

**5. Conclusion**

Through static code analysis and test coverage improvements, we achieved:  
 **Pylint score of 10/10** (code quality issues resolved)  
 **100% test coverage** (ensuring all code paths are tested)

This ensures that the triangle\_classifier.py program is clean, efficient, and fully tested.