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.

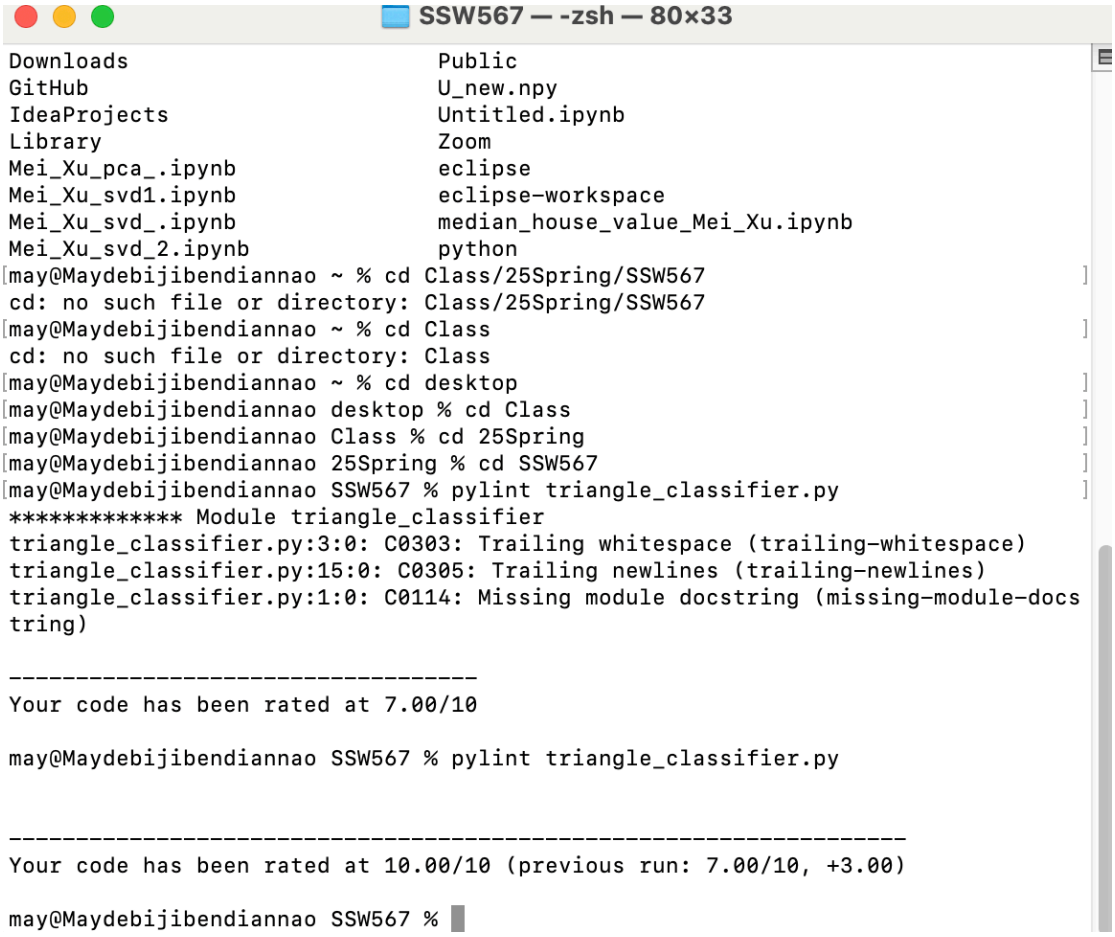
- Ensured all branches of the classify_triangle function were tested.

Final Coverage Results (After Improvements)

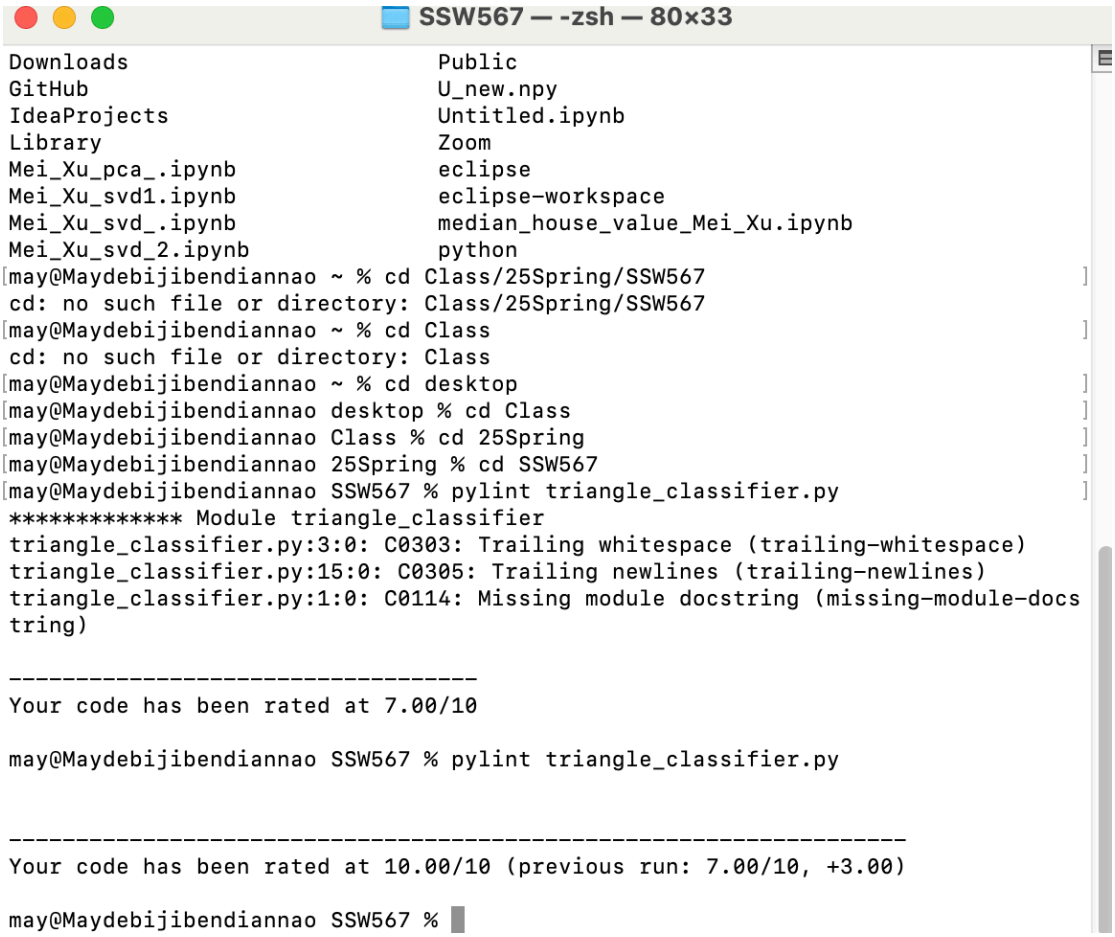
File	Statements	Missing	Coverage
triangle_classifier.py	10	0	100%
test_triangle_classifier.py	12	0	100%
Total	22	0	100%

4. Screenshots of Analysis Results

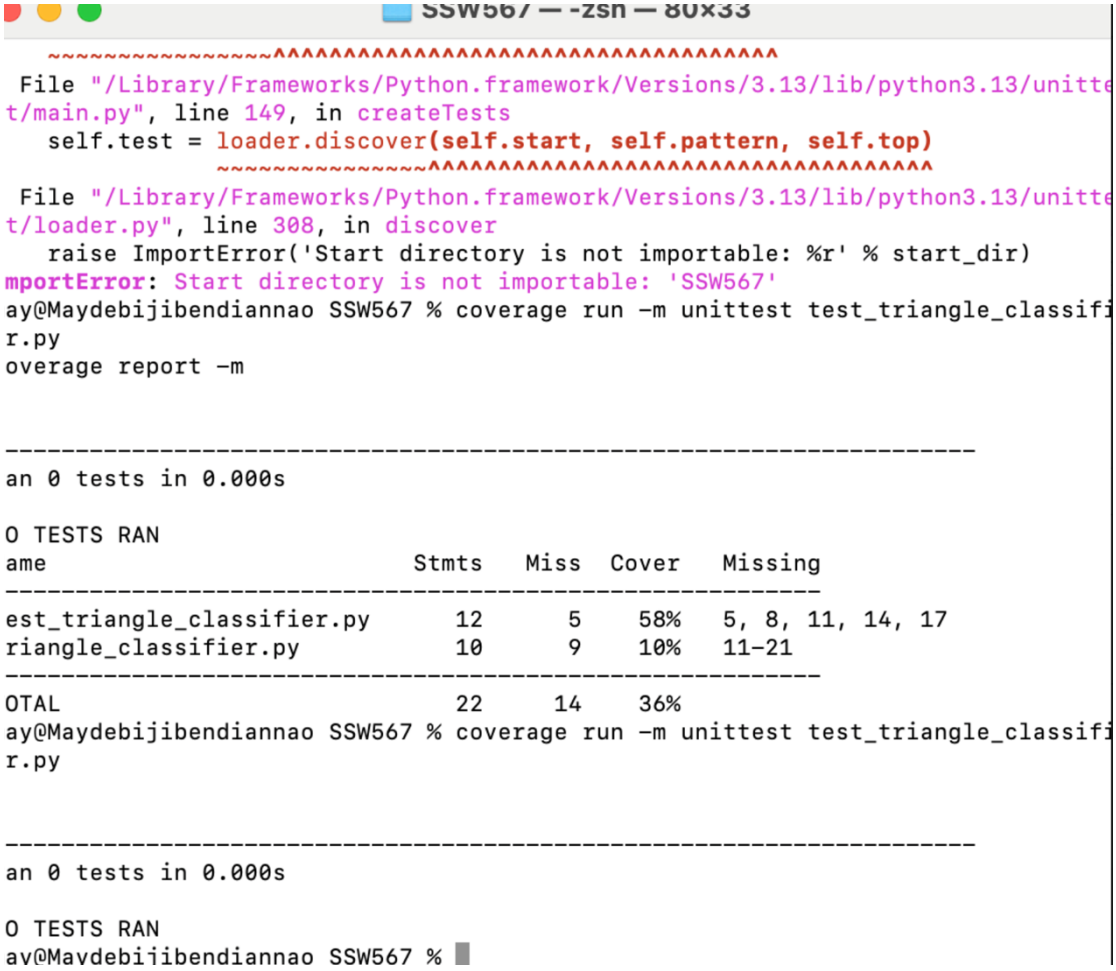
Before Fixing Errors (Pylint Score: 7.00/10)



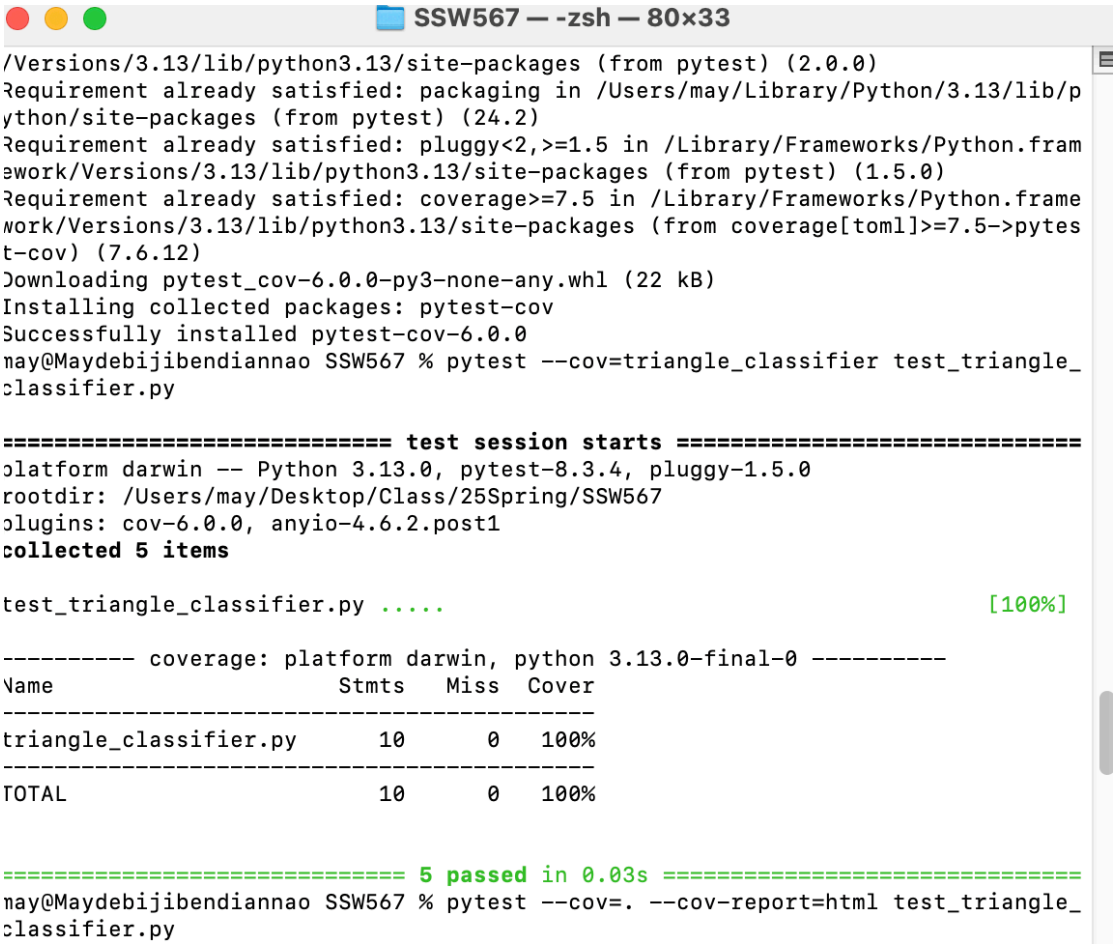
After Fixing Errors (Pylint Score: 10.00/10)



Initial Coverage Report (36%)



Final Coverage Report (100%)



```
SSW567 — -zsh — 80×33
/Versions/3.13/lib/python3.13/site-packages (from pytest) (2.0.0)
Requirement already satisfied: packaging in /Users/may/Library/Python/3.13/lib/p
ython/site-packages (from pytest) (24.2)
Requirement already satisfied: pluggy<2,>=1.5 in /Library/Frameworks/Python.fram
ework/Versions/3.13/lib/python3.13/site-packages (from pytest) (1.5.0)
Requirement already satisfied: coverage>=7.5 in /Library/Frameworks/Python.frame
work/Versions/3.13/lib/python3.13/site-packages (from coverage[toml]>=7.5->pytes
t-cov) (7.6.12)
Downloading pytest_cov-6.0.0-py3-none-any.whl (22 kB)
Installing collected packages: pytest-cov
Successfully installed pytest-cov-6.0.0
may@Maydebijibendiannao SSW567 % pytest --cov=triangle_classifier test_triangle_
classifier.py

============================= test session starts ==============================
platform darwin -- Python 3.13.0, pytest-8.3.4, pluggy-1.5.0
rootdir: /Users/may/Desktop/Class/25Spring/SSW567
plugins: cov-6.0.0, anyio-4.6.2.post1
collected 5 items

test_triangle_classifier.py .....                                        [100%]

---------- coverage: platform darwin, python 3.13.0-final-0 ----------
Name                       Stmts   Miss  Cover
----------------------------------------------
triangle_classifier.py        10      0   100%
----------------------------------------------
TOTAL                         10      0   100%


============================== 5 passed in 0.03s ===============================
may@Maydebijibendiannao SSW567 % pytest --cov=. --cov-report=html test_triangle_
classifier.py
```

Coverage report: 100%

Files | Functions | Classes

coverage.py v7.6.12, created at 2025-02-27 19:19 -0500

File ▲	statements	missing	excluded	coverage
test_triangle_classifier.py	12	0	0	100%
triangle_classifier.py	10	0	0	100%
Total	**22**	**0**	**0**	**100%**

coverage.py v7.6.12, created at 2025-02-27 19:19 -0500

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.