

Code Similarity Analysis Report

Professional Code Comparison & Analysis

CODE SIMILARITY ANALYSIS REPORT

Files Compared:

- code_1.py
- code_3.py

Overall Similarity: 74.9%

Report Generated: September 14, 2025 at 11:36:25

Analysis Category: High

Code Similarity Analysis Report

Professional Code Comparison & Analysis

Executive Summary

Project: Code Similarity Analysis
Date/Time: September 14, 2025 at 11:36:25
Primary Method: DIFFLIB
Threshold: 70%

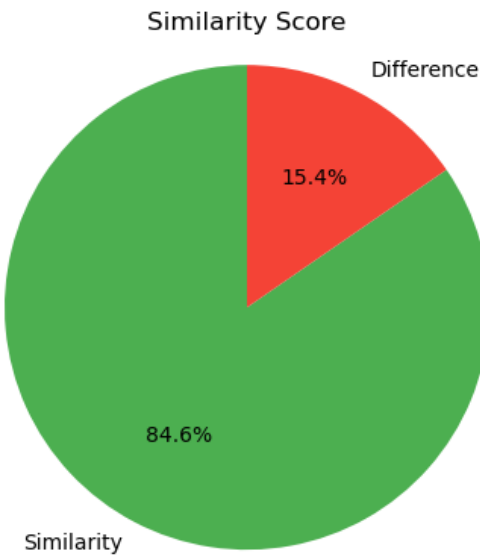
Algorithm Results Summary

Algorithm	Score (%)	Category	Status
Difflib (Sequence Matching)	84.6%	Very High	! High
TF-IDF Cosine Similarity	67.4%	High	* Normal
AST (Abstract Syntax Tree)	97.6%	Very High	! High
Jaccard Similarity	50.0%	Medium	* Normal
AVERAGE SIMILARITY	74.9%	High	! High

Similarity Visualizations

The following charts provide visual representation of the similarity analysis:

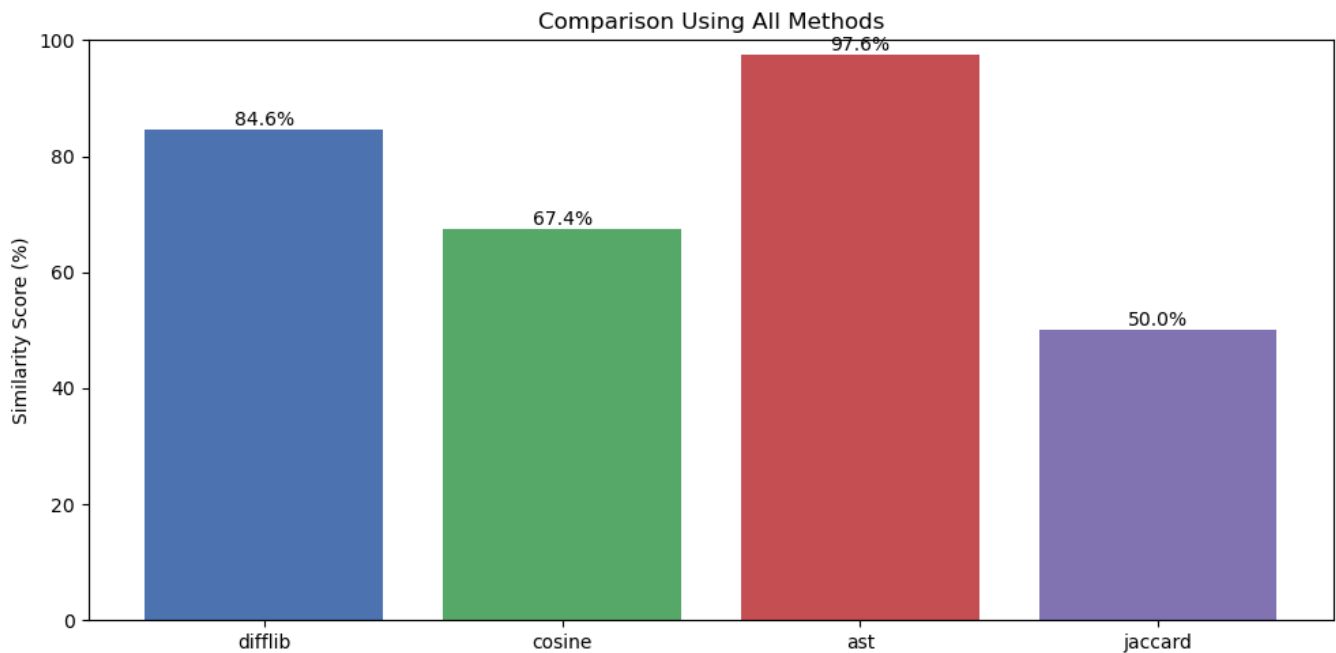
Selected Algorithm (DIFFLIB) Result:



Comprehensive Multi-Algorithm Comparison:

Code Similarity Analysis Report

Professional Code Comparison & Analysis



Detailed Algorithm Analysis

Diffliib (Sequence Matching): 84.6%

Compares code as sequences of characters, identifying longest common subsequences. Effective for detecting exact matches and minor modifications.

Result: Very High similarity detected. Strong evidence of code similarity.

TF-IDF Cosine Similarity: 67.4%

Converts code to TF-IDF vectors and calculates cosine similarity. Good for detecting structural similarities regardless of variable names.

Result: High similarity detected. Moderate similarity detected.

AST (Abstract Syntax Tree): 97.6%

Analyzes the syntactic structure of code by comparing Abstract Syntax Trees. Most effective for detecting structural plagiarism.

Result: Very High similarity detected. Strong evidence of code similarity.

Jaccard Similarity: 50.0%

Measures similarity as the ratio of common tokens to total unique tokens. Useful for detecting copied code with minor additions.

Result: Medium similarity detected. Some similarities found.

Professional Analysis & Interpretation

Code Similarity Analysis Report

Professional Code Comparison & Analysis

Overall Assessment: MODERATE SIMILARITY - The codes share significant common elements. High similarity detected by: Difflib (Sequence Matching), AST (Abstract Syntax Tree).

Conclusion

Based on the comprehensive analysis using multiple algorithms, the codes show HIGH similarity ($\geq 60\%$). There are substantial common elements between the implementations.

Code Similarity Analysis Report

Professional Code Comparison & Analysis

Technical Implementation Details

Preprocessing Pipeline:

1. Comment Removal: All single-line and multi-line comments removed
2. Identifier Normalization: Variables, functions, and classes renamed systematically
3. Whitespace Standardization: Consistent formatting applied
4. Language-Specific Processing: Custom handling for different programming languages

Code Sample Analysis

File 1: code_1.py

```
def add(a, b):  
    return a + b  
  
print(add(2, 3))
```

File 2: code_3.py

```
def subtract(a, b):  
    return a - b  
  
print(subtract(10, 4))
```

Key Differences Identified

```
--- file1  
+++ file2  
@@ -1,3 +1,3 @@  
def f0(p0, p1):  
-     return a + b  
-print(add(2, 3))  
+     return a - b  
+print(subtract(10, 4))
```