# What-If Analysis of Student Results

<https://claude.ai/public/artifacts/9074568d-5091-4c80-94e3-62dbef1a8346>

<https://claude.ai/public/artifacts/ab50d822-55e1-410f-83ab-f22361fc869a>

**Key Observations**

1. **Adaptive Penalty**: Students who get the "easy" version of Module 2 face a significant scoring penalty. For example, with 25 correct answers:
   * Hard Math: 580 points
   * Easy Math: 520 points (60-point penalty)
2. **Diminishing Returns**: At higher raw scores, each additional correct answer has less impact on the final score.
3. **Cascade Effect**: Questions in Module 1 have dual impact - they affect both raw score AND determine which Module 2 version the student gets.

**Key Algorithm Features**

**1. Adaptive Impact Calculation**

The algorithm considers that Module 1 questions have dual impact:

* Direct score impact from the raw score increase
* Adaptive impact from potentially changing Module 2 difficulty level

**2. Prioritization Strategy**

The algorithm prioritizes questions based on:

* **High raw score impact**: Questions that directly increase scaled score
* **Module 1 leverage**: Questions that could change adaptive path
* **Difficulty considerations**: Harder questions may have more impact at higher score ranges

**3. Comprehensive Analysis**

The system provides:

* Current score calculation
* Impact scoring for each incorrect question
* Recommendations ranked by potential improvement
* Summary insights about focus areas

**Implementation Strategy**

1. **Data Structure**: Structure your student response data to include question metadata (module, difficulty, subject)
2. **Threshold Tuning**: Adjust the Module 1 performance threshold based on your actual adaptive logic
3. **Validation**: Test with known student data to validate score calculations match your actual scoring system
4. **Extensions**: Consider adding:
   * Time-based difficulty adjustments
   * Question topic clustering for targeted recommendations
   * Multiple what-if scenarios

The algorithm accounts for the unique challenges of adaptive testing where changing one answer can have cascading effects on subsequent question difficulty and scoring penalties.

<https://claude.ai/public/artifacts/5059f5f6-c1b1-4d9d-8947-3e3ec02e2941>

**What is the Threshold?**

In my algorithm, I used a simple placeholder:

python

def determine\_module2\_difficulty(self, module1\_performance: float, threshold: float = 0.5) -> str:

return 'hard' if module1\_performance >= threshold else 'easy'

This assumes that if a student gets ≥50% correct in Module 1, they get the hard Module 2. But this is likely **not** how the actual DSAT works.

**Why This Matters for Impact Analysis**

The threshold directly affects your what-if analysis because:

1. **Boundary Effects**: Questions near the threshold have massive impact
2. **Cascade Effects**: A single Module 1 question could flip the entire adaptive path
3. **Score Penalties**: Getting "easy" Module 2 has significant scoring penalties

**Real-World Threshold Complexity**

The actual SAT adaptive logic is likely much more sophisticated:

SAT Adaptive Threshold Analysis

Code

import numpy as np from typing import List, Dict, Tuple import json class AdaptiveThresholdAnalyzer: """ Analyzes and tunes the adaptive threshold for Module 2 difficulty assignment """ def \_\_init\_\_(self): # These would

**Key Points for Threshold Tuning:**

**1. Why It's Critical**

* **Boundary students** (near 50% correct) have questions with massive impact
* **Wrong threshold** = wrong what-if analysis
* **Example**: If real threshold is 60% but you use 50%, you'll miss high-impact questions

**2. How to Find Your Actual Threshold**

**Method 1: Historical Data Analysis**

python

# Collect data: Module 1 performance → Module 2 assignment

students = [

{'module1\_correct': 12, 'module1\_total': 22, 'got\_hard\_module2': True},

{'module1\_correct': 11, 'module1\_total': 22, 'got\_hard\_module2': False},

# ... more data

]

# Find the cut-off point

threshold = find\_cutoff\_point(students)

**Method 2: A/B Testing**

* Test different thresholds
* See which predicts actual Module 2 assignments best
* Use cross-validation

**3. Threshold Complexity**

The real SAT threshold might be:

* **Raw score based**: "≥13 correct out of 22"
* **Weighted by difficulty**: Hard questions count more
* **IRT-based**: Uses item response theory
* **Percentile-based**: Top X% get hard module
* **Topic-weighted**: Some topics matter more

**4. Impact on What-If Analysis**

**Scenario A: Threshold = 50%**

* Student has 45% correct (10/22)
* Fixing 2 questions → 55% → Changes to hard module
* **Massive impact**: +60 points from avoiding easy penalty

**Scenario B: Threshold = 65%**

* Same student: 45% correct
* Fixing 2 questions → 55% → Still gets easy module
* **Small impact**: Only +20 points from raw score

**5. How to Implement**

python

# 1. Collect your actual data

def collect\_threshold\_data():

return query\_database("""

SELECT student\_id, module1\_correct, module1\_total,

module2\_difficulty\_received

FROM student\_responses

WHERE test\_date > '2024-01-01'

""")

# 2. Find optimal threshold

def find\_optimal\_threshold(data):

best\_accuracy = 0

best\_threshold = 0.5

for threshold in np.arange(0.3, 0.8, 0.05):

accuracy = calculate\_prediction\_accuracy(data, threshold)

if accuracy > best\_accuracy:

best\_accuracy = accuracy

best\_threshold = threshold

return best\_threshold

# 3. Update your what-if algorithm

analyzer = DSATWhatIfAnalyzer(scoring\_maps)

analyzer.adaptive\_threshold = find\_optimal\_threshold(your\_data)

**6. Validation**

After finding your threshold:

* Test on new student data
* Check prediction accuracy
* Monitor for any changes (SAT updates their algorithm)

The threshold is the **most important parameter** in your what-if analysis because it determines which questions have massive adaptive impact vs. just small score bumps.