

2026 AI 활용 사례 공모전 · 교육 (학습) 분야

오답 목록이 아니라 «취약 개념»을 찾습니다

AI를 활용해 취약 개념을 선별하고, 객관적인 근거로 학습 계획을 세웁니다

고등학교 수학 지도 · 2026년 8월

개인 응모 · 직장을 다니며 주말에 가르치는 학생 2명에게 실제로 적용한 기록입니다

저는 수학 교육에 관심이 많은 **평범한 직장인**입니다. 회사를 다니면서 **주말**에 고등학생들의 수학 공부를 도와주고 있습니다. 학생이 모의고사를 치고 오면 손에 남는 것은 «몇 번을 틀렸다»는 목록 한 장입니다. 그 목록은 **무엇을** 틀렸는지는 알려 주지만, **왜 틀렸는**지는 알려 주지 않습니다.

먼저, «취약 개념»이란

틀린 문제 그 자체가 아니라 **그 문제를 풀려면 미리 알고 있어야 했던 더 앞의 개념**을 말합니다. 이차함수 문제를 틀렸는데 진짜 원인은 인수분해일 수 있습니다. 이때 이차함수 문제를 다시 풀려면 학생은 그 문제 하나만 풀 줄 알게 될 뿐입니다. **이 사례는 그 «더 앞의 개념»을 찾아내는 방법에 대한 기록입니다.**

이 사례를 한 줄로

학생이 **무엇을 모르는지 객관적인 근거로 확인한 뒤**, 바로 그 개념을 제대로 익히고 **그 개념만 겨냥한 문제로 풀이 능력까지 갖추게** 하는 것이 이 프로그램이 하는 일입니다. **AI를 활용해 이 학생의 취약 개념을 확실히 선별하고, 객관적인 근거를 가지고 학습 계획을 세웁니다.** 그 선별은 제가 **AI로 만든 프로그램**이 미리 정해 둔 규칙대로 합니다 — 그래야 같은 답안지를 넣으면 언제나 같은 결과가 나오고, 왜 그런 결과가 나왔는지 한 걸음씩 되짚을 수 있기 때문입니다.

차례	쪽
① 어떤 상황에서 AI를 활용했나요	2
② 어떤 AI를 어떻게 활용했나요	3
③ 활용 결과 어떤 변화가 있었나요	4
④ 나만의 방식과 개선 포인트	5
⑤ 다른 사람도 따라 할 수 있나요	6

활용 AI 도구

Claude — 이 프로그램은 **Claude Code**로 만들었습니다. 만들어진 프로그램은 시험지 사진을 글로 옮기고 푸는 데, 그리고 결과를 학생이 읽을 문장으로 옮기는 데 **다시 Claude**를 부릅니다. 약점을 판단하는 단계는 **그 코드가 정해 둔 규칙대로 계산합니다** — 같은 답안지에서 결과가 달라지지 않게 하려는 것입니다.

적용 대상·기간

회사를 다니며 주말에 가르치는 **고1·고2 학생 2명**. 2026년 7월부터 8월까지, 모의고사 **5번 · 150문제**를 한 문제씩 기록했습니다.

개인정보

진단 결과 활용에 대해 학생에게 설명 후 동의 받았고 **성을 제외한 정보는 가렸습니다**. 학교와 지역은 어디에도 적지 않았습니다.

① 어떤 상황에서 AI를 활용했나요

■ 평균이 50점에 못 미치는데, 학생들은 문제집을 더 푼다

제가 가르치는 학생이 푼 세 번의 모의고사에서, **그 시험을 치른 전국 평균은 100점 만점에 45.2점 · 37.7점 · 45.7점**이었습니다 — 세 번 모두 절반에 못 미칩니다. 안타까운 일인데, 더 안타까운 것은 그 자리에서 학생들이 하는 일이 **문제집을 한 권 더 푸는 것**이라는 점이었습니다.

주변 학생들에게 물어보면 대답이 대체로 같았습니다. 학원 진도와 숙제에 하루가 다 가서 **자기가 무엇을 모르는지 확인해 볼 틈이 없다**는 것입니다. 모르는 자리를 그대로 둔 채 푼 문제 수만 늘리니, 시간은 쓰는데 오르는 않고 «해도 안 된다»는 말이 남습니다.

■ 그래서 제가 하는 일은 «무엇을 모르는지»를 먼저 확인하는 것입니다

제가 학생들을 돕는 방법은 하나입니다. 모의고사를 푼 결과를 분석해 **약한 개념을 확인하고**, 무엇을 모르는지 알려 준 다음 **그 자리부터 공부하게** 하는 것입니다. 목적 없이 문제집을 한 권 더 푸는 대신 **자기의 부족한 곳을 채우는 데 시간을 쓰게** 하면 같은 시간에 더 적게 풀고도 오릅니다. 오르는 것이 눈에 보이면 학생도 «해볼 만하다»고 느끼고, 그때부터 공부를 스스로 끌고 갑니다.

개념만 보지 않습니다. **난이도별로, 문제 유형별로**도 갈라 봅니다. 그래야 «남들도 다 틀리는 문제를 틀린 것»과 «그 학생만 유독 무너지는 자리»가 구분되고, 그림이 주어진 문제에서만 막히는 것처럼 **눈에 잘 보이지 않는 특성**까지 드러냅니다.

■ 그런데 채점 결과 앞에서 늘 같은 자리에 막혔습니다

«13번, 16번, 21번을 틀렸다»는 것은 알겠는데, 그래서 **이번 주에 무엇을 공부시켜야 하는지**가 나오지 않았습니다. 처음에는 생성형 AI에 오답 사진을 올리고 «이 학생의 약점을 알려 줘»라고 물었습니다. 답은 그럴듯했습니다. 그런데 세 가지가 걸렸습니다.

1. 같은 사진을 다시 올리면 **다른 약점**이 나왔습니다. 어느 쪽이 맞는지 확인할 방법이 없었습니다.
2. 근거를 물으면 «원래 자주 틀리는 유형이라서»처럼 **그 학생의 답안지에는 없는 이유**를 댔습니다.
3. 무엇보다, **맞힌 문제는 아예 보지 않았습니다**. 찍어서 맞은 것과 제대로 푼 것이 똑같이 «정답»으로 지나갔습니다.

그래서 질문을 바꿨습니다

«AI가 얼마나 잘 맞히나»가 아니라 **«AI를 어디에 어떻게 써야 그 답을 확인할 수 있나»**였습니다. 학생에게 «너는 이게 약하다»라고 말하려면, 그 말이 틀렸을 때 **어디서 틀렸는지 되짚어 고칠 수 있어야** 합니다. 물어볼 때마다 답이 달라지는 것에는 그럴 수가 없습니다.

② 어떤 AI를 어떻게 활용했나요

제가 만든 것은 학생의 답안지 한 장을 넣으면 «지금 무엇부터 공부해야 하는지»와 «그것을 어떻게 익히는지»를 한 묶음으로 내주는 프로그램입니다. 일하는 순서는 다섯 단계이고, 다섯 단계 전부를 AI로 만든 이 프로그램이 돌립니다.

프로그램을 만드는 데 한 번, 그 프로그램 안에서 또 한 번

프로그램 자체를 **Claude Code**로 만들었습니다. 저는 개발자가 아니라서, 터미널에서 AI와 대화하며 «이런 계산을 하는 코드를 짜 달라»고 부탁하고 그 결과를 확인하는 방식으로 쌓아 올렸습니다. 그러니 아래 다섯 단계는 **모두 AI가 만든 코드가 하는 일**이고, 그 코드가 돌면서 **필요한 자리에서 다시 AI를 부릅니다**. 만드는 데 한 번, 그 안에서 또 한 번 쓰는 셈입니다.

단계	하는 일	누가 하나
1	시험지 사진을 읽어 문제를 글로 옮기고, 어떤 개념을 묻는지 붙인다	AI에게 묻는다
2	채점 결과를 넣고 취약 개념 후보를 뽑는다	AI로 만든 코드
3	그 후보가 맞다면 반드시 결릴 문제로 확인 시험지를 만든다	AI로 만든 코드
4	학생이 그 시험을 풀면 후보마다 맞다 / 아니다 / 아직 모르겠다를 매긴다	AI로 만든 코드
5	그 결과를 학생이 읽을 문장으로 옮기고, 그 개념의 설명과 연습문제를 붙여 내준다	AI에게 묻는다

2~4단계에서는 AI에게 다시 묻지 않고 미리 정해 둔 규칙대로 계산합니다. AI를 아껴 쓰려는 것이 아니라, 물어볼 때마다 답이 달라지면 안 되기 때문입니다. 규칙으로 계산하면 같은 답안지를 넣었을 때 언제나 같은 결과가 나오고, 왜 그런 결과가 나왔는지 한 걸음씩 되짚을 수 있습니다. AI가 지어낸 말이 학생의 진단에 섞여 들어올 자리가 없습니다.

2단계 — 후보는 어떻게 뽑나요

틀린 문제에 붙어 있는 개념에서 출발해, 그 개념을 배우려면 먼저 알아야 하는 개념으로 거꾸로 거슬러 올라갑니다. 서로 다른 오답 여러 개가 같은 앞 개념에서 만나면 그것을 후보로 올립니다. 이렇게 하면 그 시험이 직접 묻지 않은 개념도 걸립니다.

확인에서 끝내지 않습니다 — 그 개념을 익히는 데까지 갑니다

무엇을 모르는지 아는 것만으로는 점수가 오르지 않습니다. 그래서 마지막 단계에서 학생에게 나가는 종이가 세 가지입니다 — 이번 시험을 읽어 주는 **시험 분석 한 장**, 약한 개념이 맞는지 가려 줄 **확인 시험지**, 그리고 **학습 설계서**입니다. 설계서에는 이번에 공부할 개념과 «왜 이 개념으로 봤는지», **그 개념만 겨냥한 연습문제와 답**이 함께 실립니다. 연습문제는 기출을 옮긴 것이 아니라 미리 만들어 둔 문제 가운데 그 개념에 맞는 것을 프로그램이 골라 붙인 것입니다. 그 문제들도 AI가 만들었지만 **답이 맞는지는 사람이 아니라 수식 계산기가 따로 확인해 둔 것**이라, 틀린 답이 학생에게 그대로 가지 않습니다.

이 종이들을 학생과 함께 놓고 이야기한 다음 **무엇부터 할지는 같이 정합니다**. 프로그램이 내놓은 순서가 늘 그대로 가는 것은 아닙니다.

학생이 겪는 것은 이 순서입니다

무엇을 모르는지 근거를 가지고 확인한다 → 그 개념을 제대로 익힌다 → 그 개념만 나오는 문제를 풀어 낸다. 세 걸음이 종이 한 묶음 안에서 이어지기 때문에, 학생은 «약점을 들었다»에서 끝나지 않고 그 자리를 실제로 넘어서는 경험을 합니다. 첨부 ③의 4~5쪽이 그 실물입니다.

③ 활용 결과 어떤 변화가 있었나요

공부를 대하는 태도가 달라졌습니다

어떤 부분이 약한지 **근거를 가지고** 알려 주니, 기계적으로 문제집을 푸는 대신 자기가 무엇을 모르는지 먼저 묻고 그 자리부터 채우려고 합니다. 공부의 목표가 분명해지면서 **스스로 학습을 끌고 가려는 의욕**이 눈에 띄게 올라왔습니다.

가장 큰 변화는 **막혀 있던 개념을 돌파했을 때** 나옵니다. 늘 틀리던 개념을 확인 시험에서 맞히고 다음 시험에서도 맞혔을 때, 그 성취감이 «수학은 어렵지만 한 과목»이라는 생각을 걷어 내고 재미를 알게 합니다. 그 재미가 다음 공부를 끌고 갑니다. 이제는 자기의 또 다른 약한 개념을 찾아보려고 문제를 더 적극적으로 풀고, **틀린 문제를 감추지 않습니다**. 그것이 무엇을 공부할지 알려 주는 재료라는 것을 알기 때문입니다. 이런 변화가 가능했던 것은 학생과 함께 놓고 이야기할 **객관적인 자료**가 생겼기 때문입니다.

이런 사례가 있었습니다 — 18문제를 맞혔는데, 제대로 푼 것은 11문제

황○○ 학생(고1)이 2026년 6월 모의고사에서 30문제 중 18문제를 맞혀 53점을 받았습니다. 저는 채점할 때 맞았는지 틀렸지만 적지 않고 **어떻게 그 답에 이르렀는지**를 다섯 가지 중에서 함께 골라 적습니다 — 제대로 풀었다 / 반만 알고 맞혔다 / 실수했다 / 풀다 막혔다 / 손도 못 댔다. **맞힌 문제에도 붙입니다**. 그렇게 적고 보니 이렇게 같았습니다.

채점 결과	문제 수
맞힘	18
↳ 제대로 풀었다	11
↳ 반만 알고 맞혔다	5
↳ 실수했는데 맞았다	1
↳ 이유를 기록하지 못한 것	1
틀림	12
↳ 풀다 막혔다	3
↳ 손도 못 댔다	9

오답 목록만 봤다면 **여섯 문제가 «아는 것»으로 그냥 넘어갔을** 겁니다. 반만 알고 맞힌 다섯 문제와 실수로 맞은 한 문제는, 다음 시험에서 그대로 오답이 될 자리입니다. 반대로 «손도 못 댔다»가 아홉 문제라는 것은 이 학생에게 지금 필요한 것이 **실수를 줄이는 연습이 아니라는** 뜻입니다.

그래서 어떻게 됐을까요. 프로그램은 이 여섯 문제까지 근거에 넣어 약한 개념 후보를 세우고, 그 후보가 맞다면 반드시 걸릴 **확인 시험지 60문제**를 만들었습니다. 학생이 그 시험에서 21문제를 틀리면서, 공부할 개념이 **네 개**로 좁혀졌습니다 — 나머지정리·인수정리, 인수분해, 이차방정식, 이차함수와 방정식·부등식. 이 넷이 **학습 설계서**가 되어 그 개념의 설명과 연습문제와 함께 학생에게 나갔습니다. 6월 모의고사의 오답 목록만 봤다면 나오지 않았을 목록입니다. 그 실물이 첨부 ③의 4~5쪽에 있습니다.

두 시험의 점수를 나란히 놓으면 오히려 잘못 읽습니다

같은 학생이 2025년 9월 기출도 풀었는데 그쪽은 64점입니다. 64점과 53점을 나란히 놓으면 «떨어졌다»고 읽게 되지만, **두 시험은 난이도가 다릅니다** — 그 시험을 본 전국 학생의 평균이 각각 45.2점과 37.7점이었습니다. 등급으로 보면 **둘 다 3등급**으로 같습니다. 그래서 이 프로그램은 점수 대신, **그 문제를 전국에서 몇 %가 틀렸는지**를 함께 넣어 계산합니다. 첨부 ③의 2쪽이 학생에게 그렇게 읽어 준 종이입니다.

④ 나만의 방식과 개선 포인트

제가 가장 신경 쓴 것은 **공부 시간의 효율**입니다. 늘 시간에 쫓기는 학생들에게는 무엇을 공부할지 고르고 **거기에만 집중하게** 하는 것이 중요합니다. 이 프로그램은 근거를 가지고 그 결정을 돕습니다.

1. **틀린 문제를 전부 다시 풀리지 않습니다.** 여러 문제에 공통으로 걸려 있는 **하나의 앞선 개념**을 찾아 거기부터 공부하게 합니다. 그 개념 하나를 메우면 여러 문제가 한꺼번에 풀리기 때문에, 같은 시간을 써도 메워지는 범위가 훨씬 넓습니다.
2. **짐작한 것을 그대로 믿지 않고 반드시 확인합니다.** 그 개념을 정말 모른다면 틀릴 수밖에 없는 문제를 골라 내고, 학생이 다 맞히면 그 개념은 공부할 목록에서 **지웁니다**. 이미 아는 것을 다시 공부하는 시간이 학생에게 가장 아까운 시간이기 때문입니다. 근거가 모자라면 결론을 내지 않고 **«아직 모르겠다»**로 남겨 둔 뒤 다음 시험에서 다시 봅니다.
3. **맞은 문제도 그냥 넘기지 않습니다.** 반만 알고 맞힌 문제나 짚어서 맞힌 문제는 다음 시험에서 그대로 틀릴 자리라서 공부할 목록에 함께 올립니다. 오답 목록만 보는 방식으로는 **절대 보이지 않는 부분**입니다.
4. **어려운 문제를 틀린 것과 쉬운 문제를 틀린 것을 같은 무게로 보지 않습니다.** 전국에서 몇 %가 틀렸는지와 배점을 함께 보기 때문에, 남들도 다 틀리는 문제에서는 약점을 찾지 않고 **그 학생 수준에서 유독 부족한 곳만** 골라냅니다.
5. **예상 결과를 미리 적어 두고 나중에 실제와 맞춰 봅니다.** «이 학생은 이 문제를 맞힐 것이다»를 미리 적어 두었다가, 그 시험을 실제로 치면 채점합니다. 지금까지 90문제를 채점했고 아무 근거 없이 찍는 것보다 나았지만 **표본이 작아 결론이라고 말하기는 어렵습니다**. 자랑하려고 둔 장치가 아닙니다. **나아지는지도 모르는 채 복잡해지지만 하는 것을 막으려고** 둔 장치이고, 학생과 함께 볼 좋은 자료도 됩니다.

같은 답안지 한 장인데, 무엇이 달라지나

같은 답안지에서	오답 목록만 볼 때	이 프로그램
공부할 것을 고르는 기준	틀린 문제 그 자체	여러 문제에 공통으로 걸린 더 앞의 개념
맞힌 문제	그냥 넘어간다	반만 알고 맞힌 것·짚어서 맞힌 것을 목록에 올린다
짐작이 틀렸을 때	틀렸는지 알 수 없다	확인 시험에서 다 맞히면 목록에서 지운다
남들도 다 틀린 문제	내 약점으로 센다	오답률·배점을 함께 봐서 약점으로 세지 않는다
근거가 모자랄 때	그래도 결론을 낸다	«아직 모르겠다»로 남기고 다음 시험에서 다시 본다

제가 잘못 적었을 때는 어떻게 되나

③에서 말씀드린 대로, 저는 채점할 때 맞았는지 틀렸는지만 적지 않고 문제마다 **어떻게 그 답에 이르렀는지**를 함께 골라 적습니다 — 제대로 풀었다 / 반만 알고 맞혔다 / 실수했다 / 풀다 막혔다 / 손도 못 댔다. 위의 판단은 전부 이 기록에서 출발합니다. 그런데 이 칸은 **제가 학생을 보고 짐작해 적는 것이라 틀릴 수 있습니다**.

그래서 프로그램은 이 칸을 하나씩 **다른 값으로 바꿔 처음부터 다시 계산해 봅니다**. 바뀌도 공부할 개념 목록이 그대로인 문제는 제가 잘못 적었더라도 상관이 없고, 목록이 달라지는 문제만 학생에게 다시 물어보면 됩니다. 세 번의 시험 90문제를 그렇게 전부 흔들어 보니 **다섯 문제**였습니다 — 아흔 문제를 다시 물어볼 필요가 없습니다.

어느 과목까지 될까요

이 방법이 서는 조건은 하나입니다 — **«먼저 알아야 하는 개념»의 사슬이 있는 과목**이어야 합니다. 그 사슬이 있으면 틀린 자리에서 **한 걸음 뒤로 물러설 곳**이 생깁니다. 수학이 가장 잘 맞고, 과학과 영어 문법도 **같은 순서가 설 것으로 봅니다**. 반대로 외워야 아는 것이 대부분인 과목은 뒤로 물러설 자리가 없어 이득이 작습니다. **다만 제가 실제로 해 본 것은 수학뿐입니다**.

⑤ 다른 사람도 따라 할 수 있나요

따라 하실 것은 프로그램이 아니라 순서입니다

이 순서는 **채팅창 하나와 학생의 답안지 한 장만** 있으면 그대로 밟을 수 있습니다. 앞에서 말한 시간 효율은 복잡한 코드 없이도 얻을 수 있고, **제 프로그램을 똑같이 만드실 필요는 없습니다**.

1. 틀린 문제를 사진으로 올려 어떤 개념을 묻는 문제인지 적게 합니다.
2. 그 개념들을 배우려면 **먼저 알아야 하는 것**이 무엇인지 표로 만들게 합니다.
3. 여러 오답에서 **겉치는 앞 개념**을 사람이 고릅니다. AI에게 고르게 하지 않습니다.
4. 그 개념만 묻는 **쉬운 문제 세 개**를 만들게 해서 학생에게 풀립니다.
5. 다 맞으면 그 후보를 **버립니다**. 여기서 버리지 않으면 앞의 넷이 소용없습니다.

핵심은 3번과 5번입니다

«이 학생이 무엇에 약한지»를 AI에게 **바로 묻는 대신**, AI로 **확인할 문제를 만들어 근거를 모으십시오**. 제가 만든 프로그램에서는 **재료를 만드는 일도 고르는 일도 모두 코드가 합니다** — 코드가 없으면 **그 재료를 놓고 고르는 일만 사람이 하면 됩니다**. AI를 적게 쓰라는 말이 아닙니다 — 1·2·4번은 AI 없이는 할 수 없는 일입니다. 자세한 절차는 첨부 ②에 있습니다.

제가 쌓은 파이썬 파일은 **291개 · 약 11만 줄**입니다. 그런데 그것은 이 순서를 **매주 빠르게** 돌리려고 쌓은 것이지, **순서 자체에 필요한 것이 아닙니다**. 학생 한 명을 놓고 한 번 해 보는 데는 채팅창 하나면 됩니다.

이 순서는 **학생 본인도, 학부모도, 선생님도** 밟을 수 있습니다. 프로그램이 있어야 하는 일이 아니라 순서가 있어야 하는 일이기 때문입니다. 학생들은 수학 학습에 엄청난 시간과 비용을 들이고 있지만 그 노력은 결과로 나타나지 않는 것 같습니다. **AI를 교육에 이렇게 쓰면 그 시간을 덜 흘려보낼 수 있다고 생각합니다**.

열심히 하는데도 점수가 오르지 않아 지친 학생들이 많습니다. 노력이 부족해서가 아니라 **어디를 채워야 하는지 몰라서**인 경우가 많습니다. 무엇이 비어 있는지 보이면 학생들은 스스로 채워 나갑니다. 제가 만난 학생들이 그랬습니다. **이 시도가 지금 막막한 학생들에게 조금이나마 도움이 되면 좋겠습니다**.

개인정보 · 자료 출처 · AI 활용 고지

개인정보 — 진단 결과 활용에 대해 학생에게 설명 후 동의 받았고, 성을 제외한 정보는 가렸습니다. 학교와 지역은 어디에도 적지 않았습니다. 진단 자료는 코칭이 끝나면 파기합니다.

자료 출처 — 평가원 및 교육청 모의고사와 그 분석 자료를 참고했습니다.

AI 활용 — 프로그램은 Claude Code로 만들었고, 그 프로그램 안에서 Claude를 문제 전사·풀이 생성과 문서 서술에 사용했습니다. 약점을 판단하는 단계는 그렇게 만든 코드가 정해 둔 규칙대로 계산합니다 — 같은 답안지에서 결과가 달라지지 않게 하기 위해서입니다. AI가 만든 문장은 사람이 모두 확인한 뒤 내보냈습니다.