2025-KELI-004

교과목명

인공지능 기초

교수명

김 병 정

대학명

한림대학교

전공명

소프트웨어학부 스마트IoT

AI 활용 수업 사례

01

AI 활용 수업 계획서

AI 활용 수업 계획서

한림대학교

교과목명	인공지능 기초		
담당교원	김병정	**개설학과 (전공)**	소프트웨어학부 스마트IoT
개설학기	2025-1	**이수구분**	전공선택
학년	2학년	**학점**	3학점
수강인원	30	**수업유형**	하이브리드
활용 AI 도구 (코스웨어 포함)	ChatGPT(4.0, 5.0, 5.2), Gemini(Gemini Pro 3 포함), AI 튜터, AI 조교		

수업개요	본 교과목 「인공지능 기초」는 소프트웨어학부 전공 교과목으로, 초급 프로그래밍 학습자가 코드 작성, 디버깅, 개념 이해 과정에서 겪는 어려움을 줄이고 문제 해결 역량과 자기주도 학습 능력을 높이기 위해 AI 결합형으로 설계되었다. 기존의 프로그래밍 입문 수업은 교수자 설명과 정해진 문제 풀이 중심으로 운영되어 학습자별 이해 수준 차이에 즉각 대응하기 어렵고, 실시간 피드백에도 한계가 있었다. 이에 따라 본 교과목은 생성형 AI, AI 튜터, AI 조교를 활용하여 학습자의 질문, 코드 작성, 개념 이해, 디버깅 과정을 개별화하여 지원하고, AI를 정답 제시 도구가 아니라 사고 확장과 문제 해결을 돕는 학습 파트너로 활용하도록 설계하였다. 특히 파이썬 기초 문법과 자료구조, 클래스, Iterator 등 프로그래밍의 핵심 요소에서 출발해 선형회귀, 분류, SVM, 결정트리, 신경망 등 기초 머신러닝 주제로 확장되는 전공 특성을 반영하여, 프로그래밍 사고력과 AI 모델 이해를 통합적으로 기르도록 구성하였다. 수업 운영은 이론 학습, 실습, AI 상호작용, 검증, 재구성이 순환하는 문제해결형 구조를 바탕으로 한다. 수업 전에는 교수자가 AI를 활용해 자료, 코드 예제, 퀴즈, 예상 질문 등을 준비하고, 수업 중에는 학습자가 ChatGPT, Gemini 등 생성형 AI와 AI 튜터·조교를 통해 개념 설명, 예제 탐색, 디버깅, 추가 문제 해결을 수행하며, 수업 후에는 코드 비교, 오류 분석, 복습 자료 생성과 피드백 반영을 통해 학습을 심화한다. 이 과정에서 학생은 AI의 답변을 그대로 수용하지 않고 자신의 코드와 비교 및 검토하며 타당성을 점검하고, 결과를 다시 구현하고 재현하는 훈련을 수행한다.

	이러한 구조는 즉각적 피드백 부족 문제를 보완하고 자기주도적 학습을 촉진할 뿐 아니라, 비판적 사고, 검증 습관, 윤리적 AI 활용 태도를 함께 기르게 한다는 점에서 교육적 의의가 크다. 또한 교수자에게는 반복 업무를 줄이고 심화 피드백과 개별 코칭에 집중할 수 있는 기반을 제공함으로써, AI를 교수학습 구조 혁신의 핵심 요소로 통합한 전공 교육 사례라는 의미를 갖는다.	
수업목표	**지식 (Knowledge)**	학습자는 파이썬 기초 문법과 자료구조, 문자열 처리, 정규표현식, 리스트 컴프리헨션, Iterator, class 등 프로그래밍의 기본 개념을 이해하고, 이를 바탕으로 인공지능 및 머신러닝 학습에 필요한 전산적 사고의 기초를 갖추도록 한다. 또한 선형회귀, 분류, SVM, 결정트리, 신경망 등 주요 AI·머신러닝 개념의 원리와 구조, 전처리-학습-평가의 흐름을 이해하도록 한다. 아울러 생성형 AI, AI 튜터, AI 조교의 역할과 활용 원리, AI 답변의 장점과 한계, 윤리·저작권·출처 문제를 함께 인식함으로써 전공 학습과 AI 활용을 연결하는 개념적 기반을 형성하도록 한다.
	기술 (Skill)	학습자는 프로그래밍 문제를 분석하고 코드를 작성·수정·개선하며, 실행 결과를 바탕으로 오류를 진단하고 디버깅하는 능력을 기르도록 한다. 또한 AI를 활용해 개념 설명을 요청하고 예제 코드를 비교·분석하며, 본인 코드와 AI 생성 코드를 대조하여 문제 해결 전략을 구체화하도록 한다. 나아가 전처리, 모델 학습, 평가, 결과 해석의 전 과정을 수행하고, AI가 제안한 코드나 해설을 자신의 방식으로 재구성하여 재현 가능하게 구현하는 역량을 함양하도록 한다. 이와 함께 주석 작성, 결과 요약, 비교 분석, 성능 해석, 피드백 반영을 통해 결과물을 지속적으로 보완·고도화하는 능력을 기르도록 한다.
	태도 (Attitude)	학습자는 AI를 정답 제시 도구가 아니라 자신의 이해를 점검하고 사고를 확장하는 학습 파트너로 활용하는 자기주도적 태도를 갖추도록 한다. 또한 먼저 스스로 사고한 뒤 AI의 도움을 받아 개념과 오류를 검토하고, 다시 자신의 판단으로 답변의 타당성을 확인하는 비판적 태도를 기르도록 한다. 아울러 AI 답변을 무조건 신뢰하지 않고 근거와 정확성을 점검하며, 저작권, 출처, 윤리적 활용, 과도한 의존의 문제를 인식하는 책임 있는 태도를 함양하도록 한다. 궁극적으로는 전공 학습 전반에서 스스로 질문하고 검토하며 더 나은 해결 방식을 탐색하는 성찰적 태도를 갖추도록 한다.

수업계획			
수업운영단계	**주요 내용 및 활동**	**AI 활용 방식**	**평가·산출물**
학습자 사전진단	• 학습자의 파이썬 기초 문법 이해 수준, 코드 작성 및 디버깅 경험, 실습 환경 활용 가능 여부를 진단한다. • 생성형 AI, AI 튜터, AI 조교에 대한 사전 사용 경험과 활용 인식 수준을 확인한다. • 프로그래밍 학습에서 학생이 주로 어려움을 겪는 개념, 오류 유형, 질문 방식, 자기주도 학습 수준을 파악한다. • 진단 결과를 바탕으로 수업 중 지원이 필요한 핵심 개념과 보완 영역을 도출한다.	• 기초 개념 확인용 진단 문항, 예상 질문 목록, 수준별 보충자료 초안을 AI로 생성하여 진단에 활용한다. • 학생의 AI 활용 경험을 확인하기 위한 간단한 사전 설문 문항을 AI로 보완·정리한다. • 교수자는 AI를 활용해 학습자의 주요 어려움과 빈발 질문을 예측하고, 수업 자료에 반영한다.	• 사전진단 설문지 또는 체크리스트 • 파이썬 기초 확인 결과 • AI 활용 경험 조사 결과 • 학습자 수준 분석 요약표
AI 활용 방법	• 생성형 AI, AI 튜터, AI 조교의 기능과 차이를 소개하고, 수업에서의 활용 원칙을 안내한다. • AI를 정답 제공 도구가 아니라 개념 이해, 예제 탐색, 코드 비교, 디버깅, 복습 지원 도구로 활용하도록 지도한다. • 질문을 구체적으로 작성하는 방법, AI 답변을 검증하는 방법, 본인 코드와 비교·재구성하는 방법을 실습한다. • AI 활용 시 저작권, 출처 표기, 윤리적 활용, 과도한 의존의 위험성도 함께 안내한다.	• ChatGPT, Gemini 등 생성형 AI를 활용하여 개념 설명, 예제 코드 생성, 오류 원인 분석, 추가 문제 제시를 시연한다. • AI 답변과 학생 작성 코드를 비교하고, 더 적절한 풀이를 선택·수정하는 방식으로 활용한다. • 수업 전에는 자료 제작, 수업 중에는 질의응답·디버깅, 수업 후에는 복습 자료 생성과 피드백 지원에 활용한다.	• AI 활용 가이드 안내 자료 • 프롬프트 작성 및 답변 검토 실습 결과 • AI 생성 코드와 본인 코드 비교표 • AI 활용 성찰 기록 또는 간단 보고서
토론·협업·프로젝트 수행 (해당시 작성)	• 학생 간 AI 활용 경험을 공유하고, 동일 문제에 대한 다양한 풀이 방식과 코드 구현 방법을 비교·토론한다. • AI가 제안한 예제, 디버깅 결과, 개선 방향을 바탕으로 문제 해결 접근법의 차이를 검토한다. • 필요 시 개별 또는 소규모 협업 형태로 코드 리뷰, 결과 해석, 설정 변경에 따른 성능 차이를 논의한다.	• 생성형 AI를 활용하여 추가 예제, 실습 문제, 코드 개선 방향을 도출하고 이를 토론 자료로 활용한다. • 학생이 AI에 질문한 결과와 본인 코드 결과를 함께 비교하여 더 적절한 해결 방법을 선택하도록 한다. • AI 답변을 그대로 수용하지 않고, 상호 검토와 재현 과정을 통해 타당성을 점검하도록 한다.	• 코드 비교표 또는 토론 기록지 • 문제 해결 과정 정리 보고서 • 설정 변경 및 결과 비교표 • 코드 리뷰 결과 또는 협업 활동 기록
학습지원 및 피드백	• 수업 중 학생 질문에 대한 즉각적 응답, 코드 오류 진단, 개	• 생성형 AI와 AI 조교를 활용하여 실시간 질의응	• 개별 피드백 기록

	념 보충 설명, 추가 예제 제공을 통해 학습을 지원한다. • 수업 후에는 과제·답안에 대한 핵심 오류, 개선 방향, 보완 학습 자료를 제공하여 자기주도적 복습을 돕는다. • 교수자는 AI가 처리한 기초 질문 결과를 바탕으로 심화 질문, 코드 리뷰, 개별 피드백에 집중한다.	답, 디버깅 지원, 추가 자료 추천을 수행한다. • AI 기반 코드 자동 채점, 서술형 답안 요약, 맞춤형 복습 자료 및 연습 문제 생성에 활용한다. • 영어 자료 번역·요약, 핵심 개념 재강조, 피드백 초안 생성에도 활용한다.	• AI 기반 코드 채점 결과 • 서술형 답안 요약 및 보완 의견 • 맞춤형 복습 자료·추가 연습문제
사후진단 및 성찰	• 수업 종료 후 학습자의 개념 이해도, 문제 해결 역량, AI 활용 역량의 변화를 점검한다. • AI 활용이 학습에 미친 긍정적 효과와 한계, 의존 위험, 검증 필요성에 대해 성찰하도록 한다. • 학습자는 자신의 코드 개선 경험, 오류 수정 과정, AI 활용 방식의 적절성을 되돌아보며 향후 학습 전략을 정리한다.	• AI를 활용해 학습 결과 요약, 복습 포인트 정리, 자기점검 질문 생성, 성찰 초안 작성을 지원한다. • AI 답변의 정확성, 한계, 할루시네이션 가능성을 다시 검토하며 비판적 활용 태도를 점검한다. • 교수자는 AI를 활용해 학습자 반응과 수행 결과를 종합 정리하고 후속 보완 방향을 도출한다.	• 사후진단 설문 또는 자기점검표 • 성찰일지 또는 학습 소감문 • AI 활용 자기평가서 • 수업 개선 제안 및 향후 학습 계획

AI윤리방침	본 강의에서는 생성형 AI를 학습 보조 도구로 활용할 수 있습니다. 학생은 개념 이해, 예제 탐색, 코드 비교, 오류 분석, 디버깅, 복습 자료 정리 등의 목적으로 생성형 AI를 사용할 수 있습니다. 다만 생성형 AI를 활용한 경우에는 사용한 도구명, 사용 목적, 입력한 질문 또는 프롬프트의 내용, AI가 생성한 결과의 활용 방식 등을 과제 및 보고서에 구체적으로 명시해야 합니다. AI가 생성한 답변이나 코드를 그대로 제출하는 것은 허용하지 않으며, 반드시 본인이 내용을 이해·검토·수정한 뒤 자신의 학습 결과물로 재구성해야 합니다. 또한 AI 답변의 오류 가능성을 고려하여 결과를 비판적으로 검증해야 하며, 저작권, 출처 표기, 개인정보 보호 등 윤리적 기준을 준수해야 합니다.
수업 기대효과	본 수업은 생성형 AI, AI 튜터, AI 조교를 활용하여 학습자의 개별 질문과 코드 작성, 디버깅 과정에 대한 즉각적이고 맞춤형 지원을 제공함으로써 자기주도 학습 능력과 문제 해결 역량을 높이는 것을 기대한다. 학생은 AI와의 상호작용을 통해 개념을 보다 쉽게 이해하고, 다양한 풀이 방식과 코드 구현 방법을 비교·검토하며, 오류 원인 분석과 결과 해석 능력을 향상시킬 수 있다. 또한 전처리, 학습, 평가의 흐름을 반복적으로 실습하면서 프로그래밍 사고력과 AI 기초 이해를 함께 기를 수 있다. 교수자 측면에서도 수업 자료 제작, 예제 개발, 퀴즈 생성, 피드백 제공의 효율성이 높아져 보다 심화된 상호작용과 개별 코칭에 집중할 수 있으며, 궁극적으로는 AI를 비판적이고 책임 있게 활용하는 학습 태도 형성에 기여할 수 있다.

02

AI 활용 수업 주차별 세부 계획서

AI 활용 수업 주차별 세부 계획서

주차	강의주제	세부내용	수업방식	AI활용방법	과제·산출물
1	수업 개요 및 AI 결합형 학습 이해	• 교과목 운영 방식 및 평가 체계 이해 • 생성형 AI·AI 튜터·AI 조교의 기능 소개 • AI 활용 학습 원칙 및 윤리 안내	강의, 오리엔테이션, 시연 중심 수업	ChatGPT, Gemini를 활용하여 예상 질문, 학습자료, 퀴즈 예시를 시연하고 AI 활용 방법을 안내	AI를 활용해 배우고 싶은 주제 탐색 및 간단한 정리문 제출
2	파이썬 실습 환경 구축과 기초 문법 실습	• 파이썬 설치 및 Colab 사용법 익히기 • 문자열 Indexing·Slicing 실습 • 리스트 변환 및 입력·출력 함수 연습	실습 중심, 예제 코드 비교 활동	AI에게 입력·출력 함수 예제 코드를 요청하고, 학생 작성 코드와 비교·분석	실습 환경 구축 확인 및 AI 생성 코드와 본인 코드 차이 분석 리포트
3	파이썬 기초문법 연습문제 풀이 1	• 문자열 함수 이해 • 정규표현식 및 리스트 컴프리헨션 실습 • 3항 연산자 활용 연습	이론 설명 후 문제풀이 및 질의응답	이해되지 않는 개념을 AI에 질문하고, AI 답변의 미흡한 부분을 재질문하며 보완	AI가 설명한 내용을 예제 코드에 주석으로 정리하여 제출
4	파이썬 기초문법 연습문제 풀이 2	• global 변수 선언 이해 • 메모리맵 개념 학습 • 정규화 개념 및 관련 문제 풀이	개념 설명, AI 기반 예제 탐색, 문제 분석	학생이 이해한 개념을 AI에 설명하고 관련 예제를 요청한 뒤, 문제 해결 과정과 오류를 점검	AI 답변과 실제 코드 실행 결과를 바탕으로 개념 이해 점검 결과 제출
5	파이썬 기초문법 연습문제 풀이 3	• Packing·UnPacking 개념 학습 • randint() 및 중복 처리 실습 • 예제 코드 분석 활동	문제해결형 실습, 코드 분석 중심 수업	AI로 Packing·UnPacking 예시 문제를 생성하고, 생성 코드의 구조와 의미를 분석	Packing·UnPacking 예시 코드 주석 보고서 및 AI 제공 코드 분석 결과 제출
6	파이썬 고급 문법 1	• Iterator 개념 이해 • 컴퓨터 설계에 필요한 논리 표현 방법 7가지 학습 • 예제 코드 비교 실습	강의와 실습 병행, 코드 비교 활동	AI가 생성한 Iterator 및 논리 표현 예시 코드를 학생 코드와 비교·검토	본인 코드 작성 결과와 AI 코드 비교표 제출

7	파이썬 고급 문법 2	• class 개념 이해 • 퍼셉트론 설계 원리 학습 • 객체지향적 코드 구조 분석	예제 문제 풀이, 코드 해석 및 보고서 작성	AI를 활용해 class 기반 구현 예시와 퍼셉트론 설계 코드를 분석	예제 문제 분석 보고서 제출
8	중간고사				
9	Linear Regression	• 선형회귀의 가정 이해 • 손실함수(MSE)와 계수 해석 • 학습·평가 지표의 의미 파악	실습 중심, 모델 학습 및 결과 해석	AI가 제시한 회귀 코드로 실습하고, 계수·예측 결과를 AI 피드백과 교차 검증	전처리-학습-평가 흐름 주석 추가 및 본인 코드 재현 결과 비교 제출
10	Classification 1 - MNIST Dataset	• MNIST 데이터 전처리 • 분류 파이프라인 구성 • 정확도·혼동행렬 해석	실습, 설정 변경 실험, 결과 비교	AI가 제안한 분류 예제를 실행하고 전처리 및 모델 설정 변화에 따른 성능 차이를 분석	AI 코드 주석 작성 및 설정 변경 후 결과 요약 제출
11	Classification 2 - Linear Models	• 선형 분류 개념 이해 • 교차엔트로피 손실 학습 • 정규화가 결정경계에 미치는 영향 분석	모델 실습, 시각화, 성능 해석	AI로 선형 모델 학습 및 시각화 코드를 구성하고, 규제 강도와 학습률 변화에 따른 과적합 여부 해석	AI 결과 재현 검증 및 튜닝 개선 근거 기록 제출
12	Classification 3 - SVM	• SVM 마진 개념 이해 • 커널 함수와 C·gamma 파라미터 학습 • 일반화 성능 비교	실습, 교차검증, 파라미터 비교 분석	AI 추천 설정으로 SVM을 학습하고, 파라미터 스윕 결과를 AI와 함께 해석	C·gamma 조합 비교 실험 및 최적 설정 선택 근거 요약 제출
13	Classification 4 - Decision Tree	• 결정트리 분할 기준 이해 • 과적합과 하이퍼파라미터 제어 • 특성 중요도 해석	트리 학습 실습, 시각화, 설정 비교	AI 제공 예제로 트리 모델을 학습하고, 깊이 제한 등 설정 변화에 따른 성능과 해석 가능성을 비교	주요 파라미터 의미 주석 정리 및 설정 개선 결과 제출
14	Classification 5 - Neural Network	• MLP 신경망 기본 구조 • 활성함수·역전파·옵티마이저 이해 • 성능-복잡도 트레이드오프 분석	신경망 실습, 구조 변경 실험, 결과 해석	AI가 제시한 베이스라인 MLP를 학습시키고 층 수, 뉴런 수, 학습률 변경에 따른 결과 비교	구조 또는 하이퍼파라미터 개선 결과와 학습 해석 보고서 제출
15	기말고사				

03

AI 활용 수업 결과보고서

AI 활용 수업 결과 보고서

수업 운영 결과	
학습자 변화	**교수자 성찰**
· 수업 전 초급 프로그래밍 학습자들은 코드 작성, 디버깅, 개념 이해 과정에서 자주 막히는 경향이 있었음 교수자 설명과 정해진 문제 풀이 중심 수업에서는 개인별 이해 수준 차이에 즉각 대응하기 어려웠음 작은 오류 하나로도 학습 흐름이 끊기거나, 개념을 들어도 실제 코드 구현으로 연결하지 못하는 경우가 많았음 실시간 피드백 부족으로 자기주도 학습 지속성이 낮고, 질문을 미루거나 포기하는 학생도 있었음 **· 수업 후 변화** 생성형 AI, AI 튜터, AI 조교를 활용하여 막히는 지점에서 즉시 질문하고 피드백을 받을 수 있게 되었음 개념 탐색, 코드 구현, 디버깅을 학생 스스로 이어가는 비율이 높아졌음 단순 암기식 학습에서 벗어나 다양한 풀이 방식과 코드 구현 방법을 비교·검토하는 학습 태도가 강화되었음 AI가 제공하는 단계적 설명, 예제 코드, 오류 원인 분석을 통해 개념 이해 속도와 문제 해결 지속성이 높아졌음 본인 코드와 AI 생성 코드를 비교하고, 결과를 재현·수정·주석화하는 과정에서 논리적 사고력과 코드 해석 능력이 향상되었음 **· 대표 반응** 질문에 즉시 답을 받을 수 있어 학습 흐름이 끊기지 않았다는 반응이 나타남	**· 잘된 점** AI를 단순 보조도구가 아니라 수업 설계와 운영 전반에 실제로 통합할 수 있었음 주차별 수업 자료, 코드 예제, 개념 요약, 퀴즈 초안, 예상 질문 도출 등에 AI를 활용함으로써 준비 효율성이 높아졌음 반복적이고 기초적인 질의응답 일부를 AI가 지원하면서 교수자는 심화 설명, 코드 리뷰, 개별 피드백에 더 집중할 수 있었음 초급 프로그래밍 수업에서 가장 큰 어려움 중 하나인 즉각적 개별 피드백 부족 문제를 일정 부분 보완할 수 있었음 **· 수업 중 실제 효과** 학생 질문에 즉각 답변을 제공하고, 오류 원인 분석과 코드 개선 방향을 제시할 수 있었음 추가 예제, 실습 문제, 설명 자료를 바로 제공할 수 있어 수업 반응성과 유연성이 높아졌음 학생들이 질문을 더 자주, 더 구체적으로 제기하게 되었고, 교수자는 오개념과 병목 지점을 더 빠르게 파악할 수 있었음 AI가 처리한 기초 질문을 바탕으로 교수자는 핵심 개념 재강조와 심화 상호작용에 집중할 수 있었음 **· 수업 후 실제 효과** 코드 자동 채점, 서술형 답안 요약, 맞춤형 복습 자료 및 추가 연습문제 제공이 가능해짐 반복 업무가 줄어들면서 교수자의 피드백 효율성이 높아졌음

어려운 용어나 개념을 쉬운 비유와 예시로 이해할 수 있어 도움이 되었다는 반응이 있었음
코드 오류 원인을 더 구조적으로 파악할 수 있었고, 디버깅 부담이 줄었다는 의견이 많았음
복습 자료, 추가 예제, 설명 자료를 바로 확보할 수 있어 학습 부담이 감소했다는 반응이 확인됨
새로운 방식의 학습 경험 자체가 흥미와 동기 유발에 도움이 되었다는 긍정적 평가가 나타남

· 변화의 근거

AI를 활용한 즉각적 피드백과 개별화된 지원에 대한 만족도가 높았음
코드 오류 탐색, 개념 이해, 복습 자료 생성 측면에서 AI 활용 효과가 확인되었음
회귀, 분류, SVM, 결정트리, 신경망 등 후반부 실습에서 AI 코드의 단순 활용을 넘어 전처리-학습-평가 흐름을 주석화하고 재현·비교하는 수준으로 확장되었음
학생이 정답 수용자에서 벗어나, 설명 검토자·코드 비교자·결과 재구성자로 이동하는 양상이 나타났음

· 한계 및 유의점

일부 학생은 AI 답변을 지나치게 신뢰하거나 빠른 정답 확인 도구처럼 활용하려는 경향을 보였음
AI 활용 역량 차이에 따라 학습 효과와 성취도 편차가 발생하였음
일부 부정확한 AI 답변은 혼란을 유발하기도 하였음
따라서 비판적 검토 태도와 윤리적 활용 습관에 대한 지속적 지도가 필요함

교수자 본인에게도 다양한 AI 도구를 실험·적용하는 과정이 AI 활용 역량 강화의 계기가 되었음

· 어려웠던 점

AI가 제시한 코드와 설명을 그대로 신뢰할 수 없어, 교수자가 직접 실행·검증·보완해야 하는 부담이 있었음
특히 코드 관련 답변은 겉보기에는 타당해 보여도 실제 오류나 설명의 불완전성이 존재하는 경우가 있었음
초기에는 AI 답변 검토와 보완에 추가 시간이 소요되어 오히려 업무 부담이 증가한 측면도 있었음
학생별 AI 활용 수준 차이가 커 동일한 도구를 사용해도 효과 차이가 발생하였음
AI 활용 윤리, 저작권, 출처 명시 등 부수적 관리 업무도 함께 요구되었음

· 종합 성찰

AI 활용은 수업 준비·운영·평가 전 과정에서 효율성과 상호작용성을 높이는 데 실질적 도움이 되었음
다만 효과는 AI 자체보다, 학생이 어떻게 질문하고 어떻게 검증하며 어떻게 자기 코드로 재구성하느냐에 따라 크게 달라졌음
앞으로는 AI 질문법, 답변 검증법, 윤리적 활용 방법에 대한 체계적 안내가 더욱 필요함
표준 AI 도구, 수업용 가이드, 교수자·학생 대상 교육 프로그램이 함께 마련될 경우 교육 효과가 더욱 안정적으로 확산될 것으로 판단됨

우수 과제 사례	본 교과목에서 확인된 우수 과제 사례는 단일한 최종 프로젝트의 성과라기보다, 주차별 AI 결합 활동 속에서 학생이 보여준 학습 방식의 변화와 결과물의 질적 향상에서 찾을 수 있었다. 실제 운영 기록을 보면 학생들은 생성형 AI를 단순히 정답을 얻기 위한 도구로 사용하는 데 머무르지 않고, 개념을 재이해하고 코드 오류를 분석하며 자기 방식으로 결과를 재구성하는 방향으로 활용하기 시작하였다. 이는 본 교과목이 지향한 AI 활용 기반 문제해결형 수업의 목적과도 맞닿아 있으며, 우수 과제의 기준 역시 ‘AI를 얼마나 많이 사용했는가’가 아니라 ‘AI를 활용하여 얼마나 깊

이 있게 이해하고 재구성했는가'에 두어야 함을 보여준다.
대표적인 사례 중 하나는 5주차 활동에서 확인되었다. 해당 활동에서는 학생이 Gemini Pro 3를 활용하되, 정답을 바로 요구하는 방식이 아니라 문제를 해결하는 데 필요한 핵심 개념이나 공식을 먼저 힌트로 받는 방식으로 접근하였다. 이는 보고서에 명시된 바와 같이 "정답보다 이해 중심 학습"을 지향한 활용 사례이며, 학생은 막히는 문제에 대해 즉각적인 피드백을 받으면서도 먼저 스스로 생각한 뒤 AI를 활용하는 순서를 유지하였다. 또한 AI에 무조건 의존할 경우 사고력 약화나 코드 학습 과정에서의 개념 혼동이 발생할 수 있다는 점을 함께 인식하고 있었으며, 수업 내용은 AI를 통해 구조화하여 요약하고 퀴즈를 통해 다시 점검하는 방식으로 메타인지 활동까지 연결하였다. 이 사례는 AI가 학습자의 사고를 대신하는 것이 아니라, 사고를 정교화하고 학습 흐름을 지속시키는 촉진 장치로 기능할 수 있음을 보여준다.
또 다른 대표 사례는 7주차와 8주차 활동에서 보다 분명하게 드러났다. 7주차 활동에서는 학생이 ChatGPT 4.0을 활용하여 문제 풀이 과정에서 처음 접하는 함수나 개념에 대해 단순한 "사용법" 수준이 아니라 동작 원리와 의미까지 이해하려는 방향으로 학습을 확장하였다. 강의 영상이나 오프라인 수업에서 충분히 이해되지 않았던 부분을 AI에 다시 질문하고, 수업에서 다룬 개념을 구현 관점에서 보완하며, 본인이 취약한 영역인 코드 구현 부분을 중심으로 AI의 도움을 받되 한 줄씩 자신의 지식과 연결하여 읽는 방식으로 활용한 점이 특징적이었다. 이는 AI 활용이 단순 복사·붙여넣기식 과제 수행으로 흐르지 않고, 전공 수업에서 중요한 이해와 구현의 연결이라는 본질을 유지하는 방향으로 작동하였음을 보여준다.
8주차 활동에서는 보다 실제적인 문제 해결 사례가 나타났다. 보고서에 따르면 학생은 데이터 분석 과제 수행 과정에서 그래프 누락, 병합 에러, 인덱스 에러와 같은 구체적 오류를 경험하였고, 이를 해결하기 위해 Google Gemini Pro 3를 정답 제시 도구가 아니라 "원인 분석 및 해결 방향 제안 도구"로 활용하였다. 교수자나 조교의 답변을 기다리는 대신 AI를 개인 튜터처럼 활용하여 즉각적인 피드백을 받았고, 동시에 AI의 할루시네이션 가능성을 고려하여 결과를 그대로 신뢰하지 않고 반복적으로 확인·검증하는 태도를 유지하였다. 첨부된 산출물 화면에서도 코드 실행과 결과 확인 과정이 제시되어 있으며, 이는 학생이 단순한 이론 이해를 넘어 실제 코드 오류를 해결하고 실행 가능한 결과물로 연결해 나갔음을 보여주는 사례라 할 수 있다. 이러한 과제는 문제 발견-AI 활용-결과 검증-자기 재구성의 흐름이 선명하게 드러난다는 점에서 우수 사례로 해석할 수 있다.
이와 같은 사례들이 지니는 의미는 분명하다. 첫째, 학생이 AI를 활용한다고 해서 곧바로 학습의 질이 높아지는 것이 아니라, 어떤 질문을 던지고 어떤 방식으로 답변을 검토하며 어떻게 자기 코드로 재구성하는가에 따라 교육적 성과가 달라진다는 점이다. 둘째, 우수 과제의 핵심은 AI 사용 자체가 아니라, AI를 매개로 개념 이해, 오류 분석, 코드 개선, 결과 해석을 더 깊이 있게 수행하였는지 여부에 있음을 확인할 수 있었다. 셋째, 본 교과목에서 나타난 우수 과제는 학생이 수동적인 수강자에서 벗어나 능동적 학습자, 다시 말해 검토자이자 재구성자로 이동하고 있음을 보여주는 증거라는 점에서 교육적으로 의미가 크다.

교육적 성과	본 교과목의 가장 핵심적인 교육적 성과는 초급 프로그래밍 학습자가 겪는 전형적인 어려움, 즉 코드 작성의 막힘, 디버깅 부담, 개념 이해의 단절, 실시간 피드백 부족 문제를 일정 부분 완화하면서 자기주도 학습과 문제 해결 역량을 실제 수업 구조 안에서 강화하였다는 점이다. 보고서의 개발 배경에서도 제시되듯이, 본 교과목은 초급 단계 학생들이 코드 작성과 디버깅, 개념 이해 과정에서 다양한 어려움을 겪는다는 문제의식에서 출발하였고, 생성형 AI, AI 튜터, AI 조교를 활용하여 즉각적이고 개별화된 지원을 제공하고자 설계되었다. 실제 운영 결과에서도 이러한 설계 방향은 학습자의 반응과 변화에서 확인되었으며, 학생들은 AI를 통해 개별 질문에 대한 즉시 응답, 맞춤형 설명, 추가 자료, 예제 코드, 오류 분석 피드백을 받음으로써 기존보다 학습 흐름을 유지하기 쉬운 환경을 경험하였다. 학습자 역량 변화 측면에서 보면, 본 교과목은 단순한 도구 사용 경험을 제공한 것이 아니라 학습 방식의 전환을 촉진하였다. 기존 수업에서는 강의자료와 교재, 코드 샘플, 판서 중심의 설명형 수업 구조 안에서 학생 개별 질문에 충분히 대응하기 어려웠고, 코드 작성 및 디버깅 역시 학생 개인의 과제 영역으로 남아 있어 실시간 피드백에 한계가 있었다. 반면 AI 결합 후에는 학생이 수업 중 생성형 AI와 AI 조교를 활용하여 질문을 즉시 해결하고, 코드 오류 원인과 개선 방향을 확인하며, 추가 예제와 실습 문제를 탐색하는 방식이 가능해졌다. 이 과정은 학생이 자신의 이해 상태를 즉각 점검하고, 개념을 다시 탐색하며, 다양한 풀이를 비교하는 기회를 제공하였다. 결과적으로 보고서에 정리된 바와 같이, 학습자는 개념 탐색, 코드 구현, 디버깅을 스스로 수행하는 비율이 높아졌고, AI의 단계적 설명과 예제 기반 피드백을 통해 논리적 사고력과 문제 해결 역량이 향상되는 양상이 나타났다. 개별화된 학습 지원이라는 측면에서도 교육적 성과는 분명하였다. AI는 학생이 질문하는 즉시 맞춤형 설명과 추가 자료를 제공하고, 학습자의 수준과 속도에 따라 적절한 연습 문제와 보충 설명을 제공하는 역할을 수행하였다. 이는 교수자 한 사람이 동일 시간 안에 모든 학생의 질문에 대응하기 어려운 초급 프로그래밍 수업의 한계를 보완해 주었다. 보고서에 따르면 학습자 반응에서도 AI를 활용한 즉각적 피드백과 개별화된 학습 지원에 대한 만족도가 높게 나타났으며, 코드 오류 탐색, 개념 이해, 복습 자료 생성 등에서 학습 부담 감소 효과가 확인되었다. 특히 학생들이 어려운 개념을 보다 쉬운 설명과 예시로 다시 접하고, 반복 질문과 심화 질의를 통해 개념을 정교화할 수 있었다는 점은 전공 입문 단계 학습자에게 매우 중요한 성과로 볼 수 있다. 전공 적용 효과 역시 중요한 성과 중 하나이다. 본 교과목은 소프트웨어학부 전공 과목으로서 파이썬 기초 문법, 자료구조, 정규표현식, 클래스, Iterator 등 프로그래밍 기초 요소를 다루는 데서 출발하여, 이후 선형회귀, MNIST 분류, 선형모델, SVM, 결정트리, 신경망 등 기초적인 머신러닝 주제로 확장되는 구조를 가지고 있다. 즉, 이 수업은 AI 도구 사용법을 가르치는 과목이 아니라, 프로그래밍 사고력과 AI 모델 이해를 실제 코드 수준에서 연결하는 전공 수업이다. AI 결합 수업 구조는 이 연결을 보다 촘촘하게 만들어 주었다. 학생들은 예제 코드를 그대로 실행하는 데 그치지 않고, 전처리-학습-평가의 흐름을 해석하고, 설정을 변경하며, 결과 차이를 비교·정리하는 방식으로 과제를 수행하였다. 이는 전공 수업의 핵심인 구현 능력, 분석 능력, 해석 능력, 재현 능력을 동시에 자극한다는 점에서 교육적 가치가 크다.

	수업 혁신성의 측면에서 보면, 본 교과목의 성과는 단순히 “AI를 사용했다”는 수준을 넘어서 있다. 보고서의 수업 전·중·후 비교에 따르면, 수업 전에는 교수자가 교재와 기존 자료를 바탕으로 수작업으로 PPT, 코드 예제, 평가 문항을 제작하였고, 수업 중에는 주로 강의 중심 수업이 운영되었으며, 수업 후에는 과제와 시험을 중심으로 학습 성취도를 확인하고 서술형 답안과 코드 과제 피드백을 직접 작성해야 했다. 그러나 AI 결합 후에는 수업 전 단계에서 생성형 AI를 활용하여 주차별 수업 자료, 코드 예제, 개념 요약, 퀴즈 문항 초안, 예상 질문을 준비할 수 있게 되었고, 수업 중에는 실시간 질의응답, 디버깅, 추가 자료 제공, 예제 추천이 가능해졌으며, 수업 후에는 자동 채점, 답안 요약, 피드백 생성, 맞춤형 복습 자료 제공까지 연계되었다. 이처럼 AI가 교수학습 과정 전체에 걸쳐 구조적으로 배치되었다는 점은 본 수업의 혁신성을 보여주는 중요한 근거이다. 교수자 측면의 성과도 작지 않았다. 보고서에는 AI 기반 자료 생성으로 수업 준비 부담이 완화되고, 반복적 기초 질의는 AI가 담당함으로써 교수자가 심화 상호작용과 개별 코칭에 집중할 수 있었다는 점이 제시되어 있다. 실제로 초급 프로그래밍 수업에서는 설명의 반복, 유사한 질문에 대한 재응답, 단순 오류 수정 등 반복 업무가 많고, 이로 인해 교수자가 보다 깊이 있는 피드백이나 개별 지도에 충분한 시간을 확보하기 어렵다. 본 교과목에서는 AI가 이러한 반복 영역을 보조하면서, 교수자는 오히려 더 중요한 설명과 심화 피드백에 집중할 수 있는 여지를 확보하였다. 또한 다양한 AI 도구를 실험하고 실제 수업에 통합하는 과정은 교수자 자신의 AI 활용 역량을 확장하는 계기가 되었다는 점에서도 교육 운영의 부가적 성과가 있었다. 확산 가능성 측면에서도 본 교과목의 성과는 의미가 있다. 즉각적 피드백 제공, 개별화 지원, 반복 업무 경감, 보충 자료 자동화, 복습 자료 제공이라는 특징은 초급 프로그래밍 교과목에만 국한되지 않는다. 기초 데이터분석, AI 입문, 컴퓨팅 사고, 전공 기초 실습 과목 등에서도 충분히 적용 가능한 구조이며, 특히 학습자의 질문 빈도가 높고, 개념-실습-오류 수정이 긴밀하게 연결되는 과목에서 효과가 클 것으로 판단된다. 따라서 본 교과목은 단일 수업의 성공 여부를 넘어, 대학 내 AI 활용 전공 수업 모델로서 참고할 수 있는 확장 가능성을 보여주었다는 점에서 의의가 있다.
한계 및 개선사항	그럼에도 불구하고 본 교과목 운영 과정에서 확인된 한계 역시 분명하였다. 가장 먼저 지적할 수 있는 부분은 AI 결과 검증의 어려움이다. 보고서에서도 교수자 경험의 부정적 측면으로 AI 답변 검증 및 보완에 추가 시간이 소요되었고, 특히 코드 관련 응답은 제한적 신뢰성을 보여 교수자의 점검이 필수적이었다고 정리되어 있다. 이는 실제 운영에서도 반복적으로 체감된 문제였다. AI가 제시하는 설명이나 코드는 문장 수준에서는 자연스럽고 설득력 있어 보일 수 있으나, 실제 실행 환경에서는 오류가 발생하거나, 개념 설명이 충분히 엄밀하지 않거나, 부분적으로 부정확한 내용을 포함하는 경우가 있었다. 따라서 교수자는 AI 생성 자료를 수업에 활용하기 전에 반드시 직접 실행해 보고, 내용과 논리 구조를 검토하고, 오개념 발생 가능성을 점검하는 절차를 거쳐야 했다. 이는 수업 초기 단계에서 교수자의 부담을 증가시키는 요인이었다. 학습자 수준 차이에 따른 AI 활용 격차도 중요한 한계로 확인되었다. 일부 학생은 AI를 개념 이해, 오류 분석, 비교 검토, 재구성의 도구로 비교적 적절하게

활용하였지만, 일부 학생은 AI를 빠른 정답 확인 도구나 단순 해설 제공 수단으로 사용하는 경향을 보였다. 이러한 차이는 결국 학습 효과와 성취도, 과제의 질적 수준 차이로 이어질 가능성이 있었다. 보고서에서도 AI 활용 역량 차이로 인한 학습 성취도·효과성 편차가 발생하였다고 지적하고 있으며, 이는 단순히 도구 접근성의 문제가 아니라 질문 능력, 검증 능력, 자기 설명 능력의 차이와도 연결된다. 따라서 향후에는 수업 초반 오리엔테이션 단계에서 AI 활용 원칙만 소개하는 데 그칠 것이 아니라, 구체적인 질문 작성법, 잘못된 질문의 예, 검증 절차, 본인 코드와 비교하는 방법 등을 보다 체계적으로 훈련할 필요가 있다.

도구 의존성의 위험 또한 무시할 수 없다. 실제 운영 사례를 보면 학생들은 AI의 즉각적 피드백에 높은 만족을 보였고, 개념 이해와 디버깅 부담이 감소하였다는 긍정적 효과도 분명하였다. 그러나 보고서의 부정적 반응에도 나타나듯, 일부 학생은 AI 답변에 의존하면서 스스로 분석·검증하는 태도가 약화되는 양상을 보였다. 이는 특히 초급 학습자에게서 더 쉽게 나타날 수 있는 문제이다. 개념이 충분히 내면화되지 않은 상태에서 AI가 제공한 코드나 설명을 그대로 받아들이게 되면, 표면적으로는 과제가 빠르게 해결되는 것처럼 보일 수 있으나 실제 학습의 깊이는 확보되지 않을 수 있다. 따라서 앞으로는 “먼저 생각하고, 다음에 질문하고, 마지막에 반드시 검증한다”는 원칙을 수업 운영 및 평가 기준과 더 분명하게 연동할 필요가 있다. 예를 들어 과제 제출 시 AI 활용 여부를 단순 표기하는 수준이 아니라, 어떤 질문을 던졌는지, 어떤 답변을 받았는지, 그중 무엇을 수용하고 무엇을 수정하였는지를 기록하도록 하면 의존성을 낮추고 성찰적 활용을 유도하는 데 도움이 될 것이다.

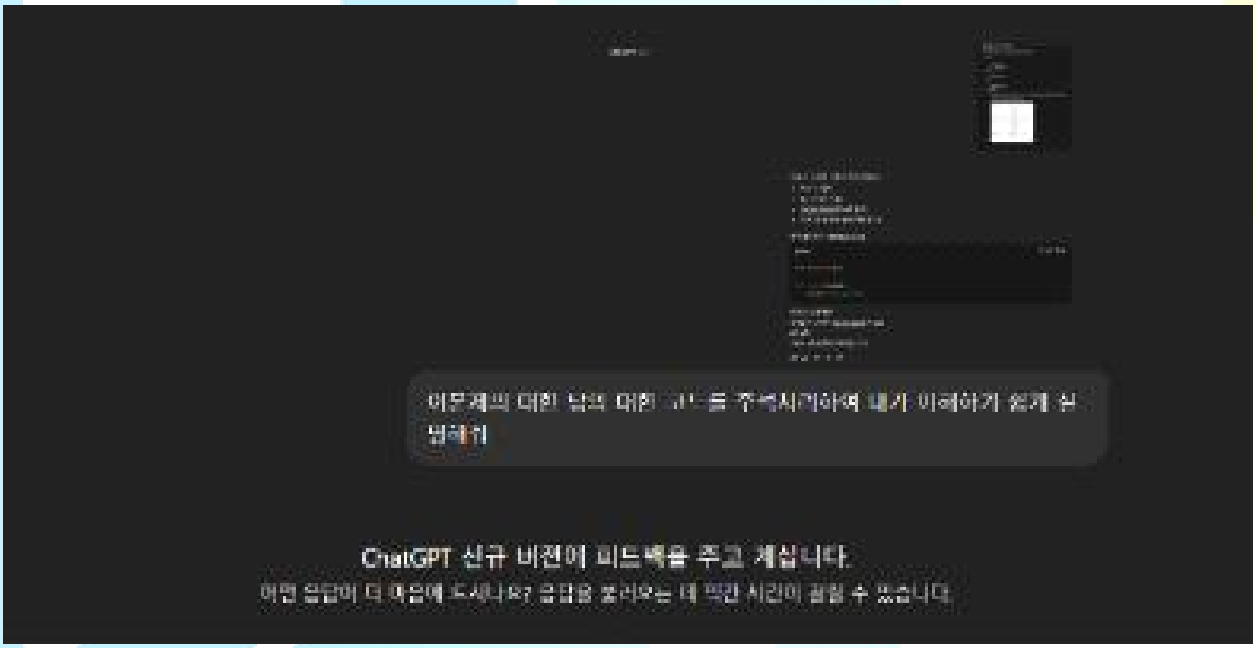
윤리 지도와 출처 관리 문제 역시 지속적인 개선 과제이다. 보고서에서는 AI 윤리, 저작권, 출처 명시 등의 부수적 관리 업무 필요성이 지적되었고, 제언 부분에서도 AI 결과물을 수업 자료로 사용할 때 출처를 명기하고 AI 생성물임을 명확히 표시해야 한다고 제시하고 있다. 이는 학생 과제에서도 마찬가지이다. 생성형 AI를 통해 얻은 설명, 코드, 요약문, 아이디어를 어느 범위까지 자기 결과물에 포함할 수 있는지, AI를 통해 번역하거나 재구성한 자료는 어떻게 표시해야 하는지, 개인정보나 외부 데이터를 다룰 때 어떤 주의가 필요한지에 대한 기준이 보다 분명해져야 한다. 향후에는 수업 차원의 윤리 서약, 과제 제출 양식 내 AI 활용 기재란, AI 활용 사례와 부적절한 사례에 대한 예시 안내 등이 함께 제공될 필요가 있다.

제도적 지원 부족 역시 현장의 한계로 드러났다. 보고서의 제언에서도 표준 AI 도구 및 플랫폼 제공, API 연계, 데이터 보안 강화, 교수자·학생 대상 AI 활용 교육 프로그램, AI 활용 데이터 분석 및 피드백 시스템이 필요하다고 명시하고 있다. 개별 교수자가 각자 적합한 도구를 탐색하고 검증하며 운영하는 방식은 초기 실험 단계에서는 가능하지만, 대학 차원의 안정적 확산을 위해서는 한계가 있다. 도구 선택의 기준, 보안 기준, 평가 기준, 윤리 기준, 수업 적용 가이드가 제도적으로 정리되어야 교수자 개인의 부담을 줄이고 교육적 품질을 일정 수준 이상 유지할 수 있다. 따라서 향후 개선 방향은 개별 수업 차원의 노력과 대학 차원의 지원 체계 구축이 병행되는 방식으로 설정되어야 한다고 본다.

시사점	본 교과목 운영은 수업 설계 측면에서 몇 가지 중요한 시사점을 제공한다. 첫째, AI는 기존 수업에 부가적으로 더해지는 보조도구가 아니라, 교수학습의 흐름을 다시 설계하게 만드는 구조적 변수로 보아야 한다는 점이다. 실제로 본 교과목은 수업 전 자료 준비, 수업 중 상호작용, 수업 후 평가와 피드백이라는 전 과정을 하나의 흐름으로 보고 AI를 배치하였다. 이러한 설계는 단순히 수업 자료 일부를 AI로 만든 것과는 질적으로 다르다. 학습자가 언제 질문하고, 어떤 지점에서 막히며, 그때 어떤 유형의 AI 지원이 필요한지를 수업 구조 속에 반영한 것이기 때문이다. 이는 앞으로 AI 활용 수업을 설계할 때도 마찬가지로 적용될 수 있는 원리라고 판단된다. 즉, AI는 수업의 '부가 기능'이 아니라 교수학습 설계의 '한 요소'로서 다루어져야 한다. 둘째, AI 활용 측면에서 중요한 것은 도구 선택 그 자체보다 활용 원리와 교육적 목적의 명확화라는 점이다. 보고서의 운영 사례를 보면 학생들은 ChatGPT, Gemini 등 서로 다른 생성형 AI를 활용하였고, 상황에 따라 어떤 도구는 설명이 더 이해하기 쉬워서 선택되고, 어떤 도구는 상대적으로 덜 활용되기도 하였다. 이는 특정 도구의 우열보다, 어떤 학습 상황에서 어떤 방식으로 활용하는지가 더 중요하다는 사실을 보여준다. 정답 제시를 요구하는 방식보다 핵심 개념의 힌트를 요청하거나, 자신의 이해를 설명한 뒤 검토를 요청하거나, 오류 원인 분석과 비교표 작성을 요청하는 방식에서 교육적 효과가 더 크다는 점이 운영 사례를 통해 확인되었다. 따라서 AI 활용 교육의 핵심은 "무엇을 쓸 것인가"보다 "어떻게 물을 것인가, 어떻게 검증할 것인가, 어떻게 자기 결과로 재구성할 것인가"를 가르치는 데 있다. 셋째, 대학 교육 차원에서 볼 때 본 사례는 전공 입문 교과목에서 AI 활용 수업이 어떤 효과를 낼 수 있는지 보여주는 실천적 모델이라는 점에서 시사적이다. 초급 프로그래밍과 같이 질문 빈도가 높고, 오류 수정과 반복 연습이 중요하며, 개념과 구현의 연결이 핵심인 교과목에서는 AI의 즉각적 피드백과 개별화 지원이 큰 장점으로 작용할 수 있다. 동시에 교수자의 반복 업무를 줄이고, 보다 심화된 상호작용과 개별 피드백에 시간을 투입할 수 있도록 만든다는 점도 중요하다. 이와 같은 특징은 컴퓨팅 기초, 데이터분석 기초, AI 입문, 공학 실습형 교과목 등에도 충분히 확장 가능하다. 따라서 본 교과목은 단지 한 과목의 사례가 아니라, AI를 활용한 전공교육 혁신 모델의 초기 형태로 이해할 수 있다. 넷째, 대학 차원의 확산을 위해서는 제도화의 조건이 반드시 함께 마련되어야 한다는 점이다. 개별 교수자가 자율적으로 AI를 수업에 도입하는 것은 시작점이 될 수 있지만, 확산 단계로 나아가기 위해서는 표준 AI 도구 제공, 데이터 보안이 확보된 환경 구축, 교수자와 학생 대상 활용 가이드와 교육 프로그램, 윤리 기준, 평가 기준, 검증 절차 등이 함께 필요하다. 보고서에서도 이러한 제도적 지원의 필요성이 제언으로 제시되어 있으며, 이는 단순한 편의 제공을 넘어 교육 품질과 신뢰성 확보를 위한 기반이라 할 수 있다. 특히 AI 활용 수업이 늘어날수록 공정성, 저작권, 출처, 개인정보 보호, 생성 결과 검증과 같은 문제가 더 자주 제기될 수 있으므로, 대학 차원에서 선제적으로 원칙을 마련할 필요가 있다. 마지막으로, 본 교과목은 AI를 활용한 수업의 본질이 '기술 도입' 그 자체가 아니라 '학습 경험의 재구성'에 있음을 보여준다. 학생이 질문을 더 자주 하게 되고, 막히는 지점에서 학습을 중단하지 않게 되며, 자신의 코드와 AI의 코드를 비교하고, 결과를 재현하고, 오류를 분석하는 방식으로 학습이 재조직될 때 비로소 AI는 교육적으로

	의미 있는 역할을 하게 된다. 교수자 역시 반복 업무에서 일정 부분 벗어나, 보다 본질적인 설명과 심화 피드백에 집중할 수 있게 된다. 이러한 변화는 단지 효율성의 문제가 아니라, 대학 교육이 어떤 방식으로 학습자의 사고를 촉진하고 성장시키는가에 관한 문제이기도 하다. 따라서 본 교과목 사례는 AI 활용 수업을 바라볼 때 단순한 도구 사용의 성패를 넘어서, 교수학습 구조와 학습자 경험의 질적 전환이라는 관점에서 해석할 필요가 있음을 시사한다.

시각자료





- 문제 해결을 위한 AI 사용
  - 학생들에게 문제지 풀이 과제 진행

- 문제지를 풀다가 막혔을 때, AI를 "정답 제시"가 아니라 조건 추가 기반의 해설/코드 설명 도구로 사용하도록 지도하여 자기주도적 이해를 유도한다.
- 문제 사진만 입력해 답을 받는 방식과, 추가 조건을 붙여 설명을 요청하는 방식을 비교한다.
- AI 답안을 그래돌 수용하지 않고, 코드 흐름 해석 및 오류 가능성 검토를 병행하는 학습 습관을 가지도록 유도한다.

1주차 활용 AI 서비스: Chat GPT 5.0

11주차

NP11-0001

역순 만들기

- 슬라이싱
- reversed 내장 함수
    - 이터레이터 반환으로 np.array로 변환 필요(복사 발생)
- np.flip(arr[, axis])
    - axis 매개변수로 축 지정 가능, 1차원에서는 axis=0과 동일

권장 순서

- 1차원 배열: 슬라이싱 > np.flip > reversed + 변환
- 다차원 배열: 특정 축만 뒤집을 경우 np.flip(arr, axis = n) 권장

- np.ndarray 에는 list의 reverse() 메서드가 없습니다.
- reversed()는 이터레이터를 반환
- 뷰(view)와 복사(copy) 동작 이해
    - 슬라이싱, np.flip은 일반적으로 뷰 방식으로 동작하여 메모리 효율적, 원본 데이터 복사하지 않고 같은 메모리 참조

Python

```python
# 방법 1
import numpy as np

arr = np.arange(0, 10) # [0..9]
arr = arr[::-1]

# 방법 2
arr = np.arange(0, 10)
rev_iter = reversed(arr) # 이터레이터 반환
arr2 = np.array(list(rev_iter))
```

- 학생이 동영상을 시청하는 과정에서 추가 설명이 제공되지 않거나, 이해가 충분하지 않은 개념이 등장할 경우, AI를 통해 해당 개념을 검색하고 보완 학습할 수 있도록 지원한다.
    - ○ 강의 영상 및 오프라인 수업, 문제지 전반에서 AI를 활용하되, 목적을 각각 다르게 설정하여 학습기록과 피드백 루프를 강화한다.
        - 강의 영상에서 이해가 안되는 부분을 정리 및 복습하고 노트에 정리한다.
        - 문제지에서 작성 코드의 리팩토링하고 더 나은 방식을 검토하고 채택한다.

2주차 활용 AI 서비스: Chat GPT 5.2 (유료 구독자)

I/O병목에 대한 설명

I/O 병목 현상 (I/O Bottleneck) 설명

I/O 병목 현상은 컴퓨터 시스템의 성능을 저하시키는 주요 원인 중 하나로, **입력(Input) 및 출력(Output)** 작업이 시스템 전체의 처리 속도를 제한하는 상황을 의미합니다.

> 마치 좁은 병의 목 부분처럼, 데이터가 오가는 통로가 느리거나 한정되어 있어서 시스템의 다른 빠른 구성 요소(CPU, 메모리)들이 데이터를 기다리느라 제 속도를 내지 못하는 현상입니다.

주요 구성 요소와 발생 원인

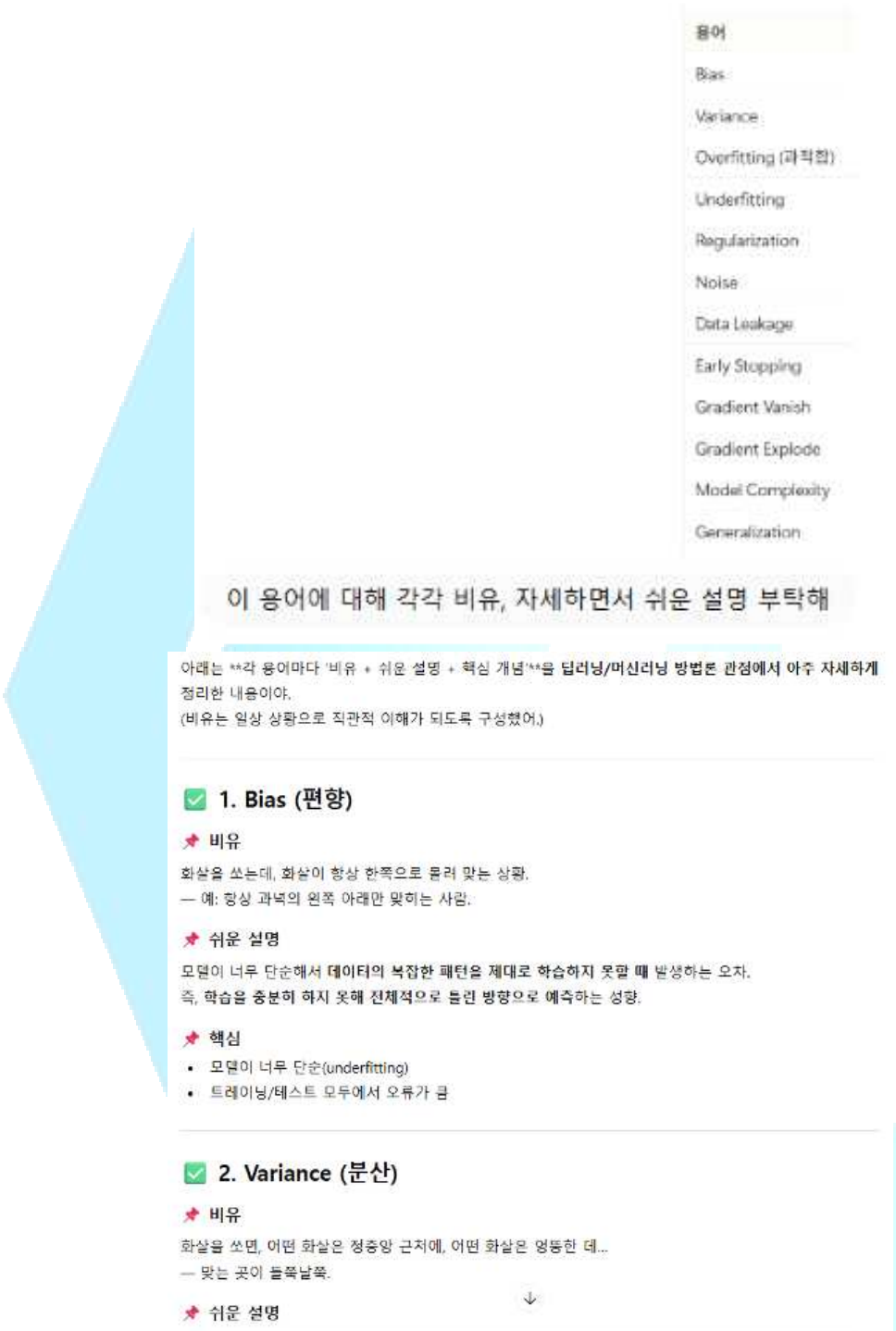
I/O 작업은 크게 디스크(저장장치) I/O와 네트워크 I/O로 나눌 수 있으며, 각각 다음과 같은 원인으로 병목 현상이 발생할 수 있습니다.

I/O 유형	발생 원인 (예시)
디스크 I/O 병목	* **느린 저장 장치:** CPU나 메모리 속도에 비해 느린 하드 디스크 드라이브(HDD) 사용 (SSD는 상대적으로 빠름). * **잦은 디스크 접근:** 데이터베이스의 잦은 읽기/쓰기 연산, 비효율적인 파일 입출력, 과도한 페이징(메모리 부족으로 디스크를 메모리처럼 사용하는 현상). * **잘못된 인덱스 설정:** 데이터베이스에서 필요한 데이터를 찾기 위해 너무 많은 디스크 I/O가 발생하는 경우.
네트워크 I/O 병목	* **대역폭 부족:** 네트워크가 처리할 수 있는 데이터 전송 용량(Bandwidth)이 부족한 경우. * **높은 지연 시간(Latency):** 원격 서버와의 통신에서 데이터가 왕복하는 데 시간이 오래 걸리는 경우. * **과도한 요청 수:** 많은 동시 접속자나 잦은 API 호출 등으로 네트워크 장비(라우터 등)의 처리 한계를 초과하는 경우.

Sheets로 내보내기

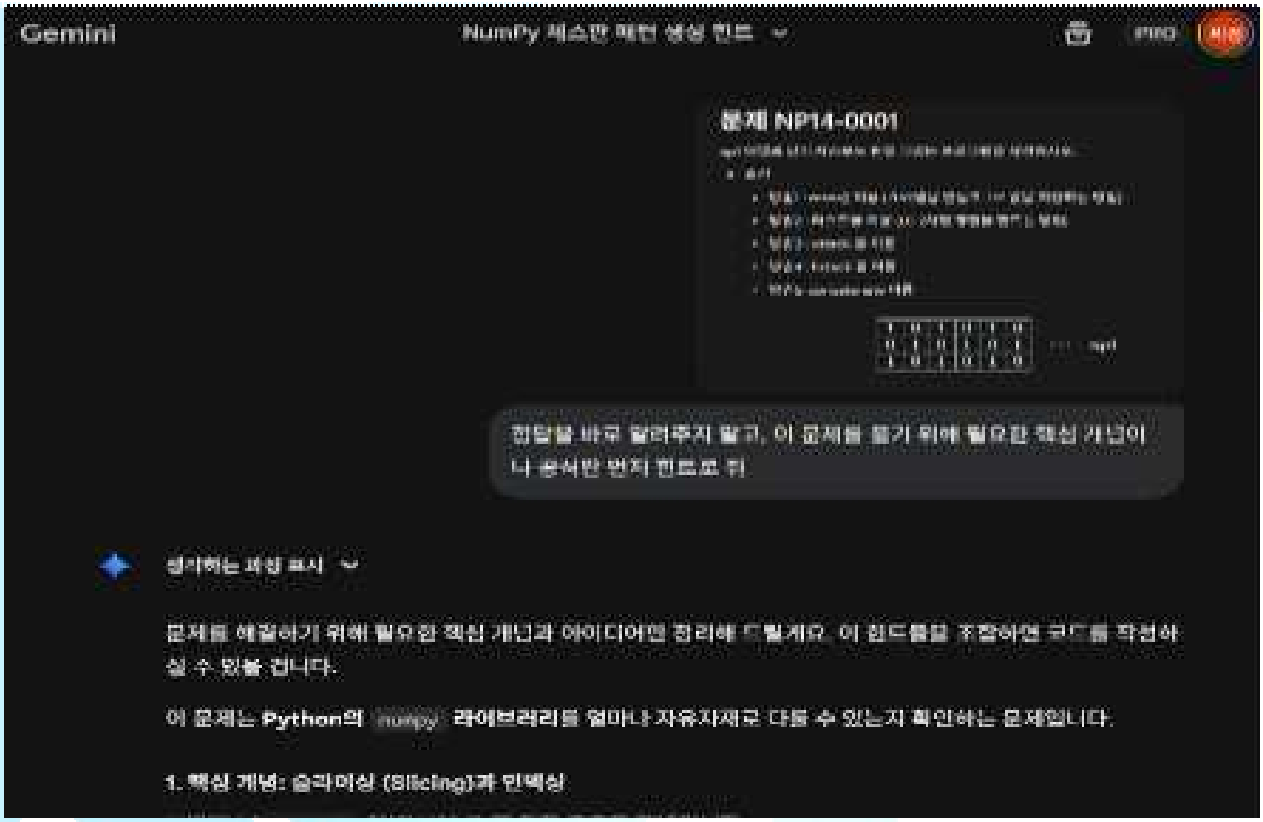
- 수업에서 문제를 풀고 오답이 나왔을 때 왜 틀렸는지에 대한 구체적인 해설을 알기 위해서 AI 사용
  - AI에게 코드의 동작 과정을 간단히 설명하며, 이해한 내용이 정확한지 검토받았다.
    - 문제 해결에 있어서 어떤 점이 잘못됐는지 본인의 생각과 결과를 설명 후 AI에게 질문
    - 요청하고 기존에 문제와 연관해서 말해줌으로서 좀 더 쉽게 공부할 수 있었습니다.
    - 오답 원인을 단순히 틀렸다고 생각하지 않고, AI를 활용해 병목 원인/개념 차이(비교표)등을 구조적으로 재학습하도록 설계한다.
    - 정보성 질문을 빠르게 해결하되, 검증 없이 맹신하지 않는 태도를 가지도록 설명한다.

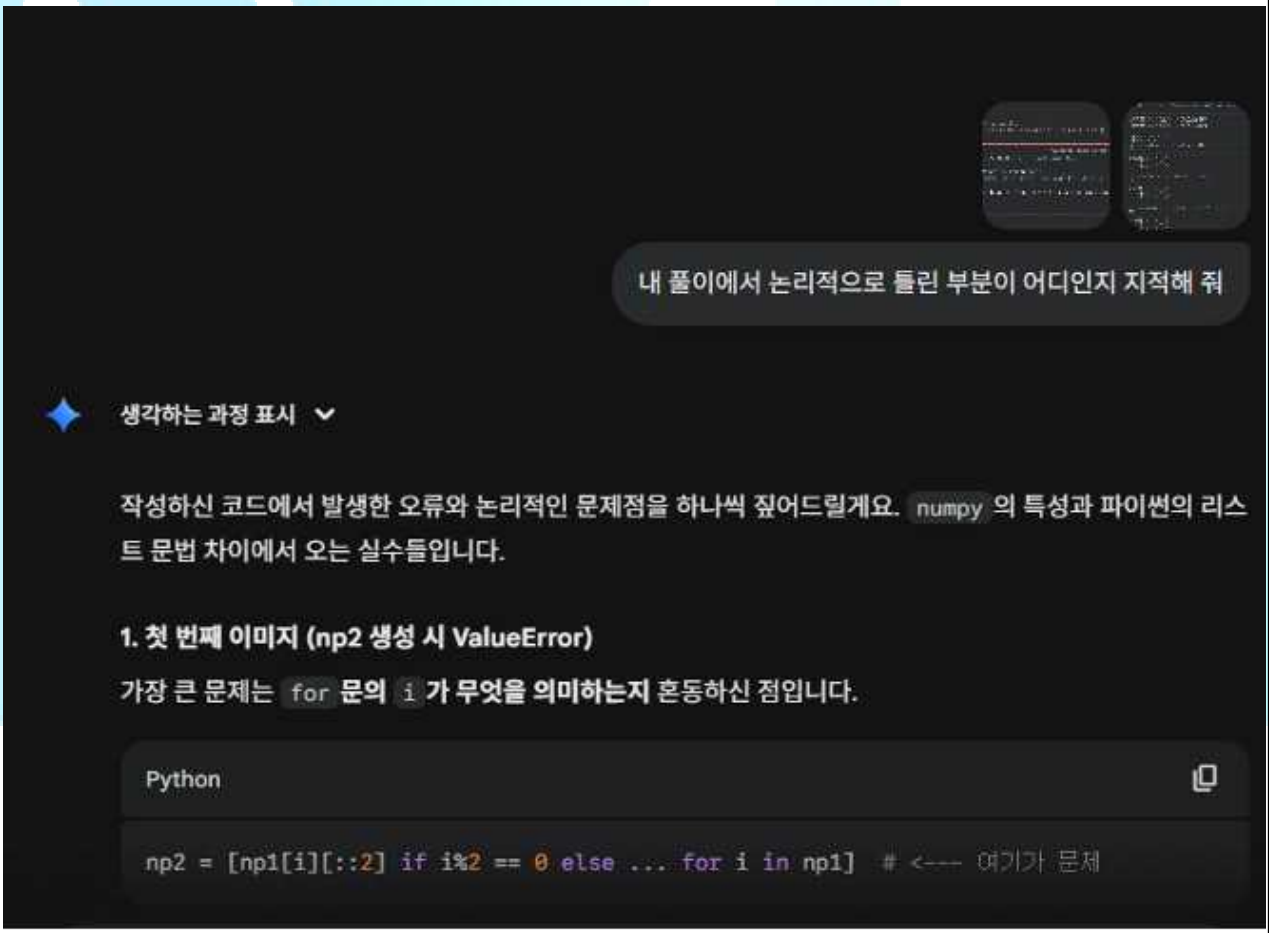
3주차 활용 AI 서비스: Gemini / chatgpt

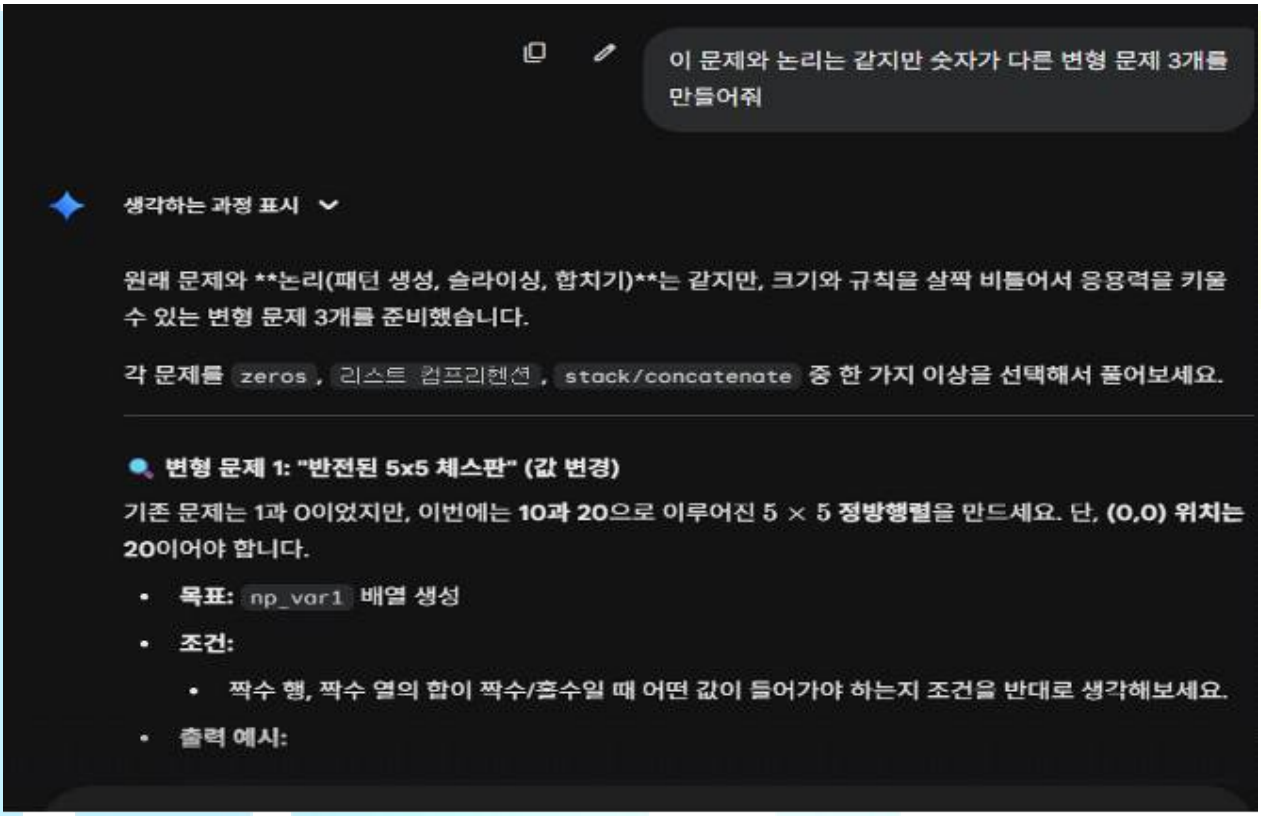


- 강의에서 다룬 용어가 어려워서 AI에게 쉽게 설명을 요청했다.
  - 우리 실생활에서 사용되는 용어를 활용해서 설명 해달라고 했다.
    - 이렇게 하면서 수업시간에 잘 이해되지 않았던 부분을 추가적으로 쉽게 알 수 있어서 좋다.
    - 문제지 풀이에서 부분적으로 막히는 지점을 AI로 해결하여 학습 속도를 높이되, 단순 검색 대비 효율을 체감했다.
    - 강의 영상은 직접 타이핑으로 학습하고, 오프라인 수업에서는 필수 용어/핵심 개념 정리와 요약을 중심으로 AI를 제한적으로 사용했다.

4주차 활용 AI 서비스: Chat GPT 4.0


Gemini
NumPy 제스판 패턴 생성 힌트
문제 NP14-0001
생각하는 과정 표시
문제를 해결하기 위해 필요한 핵심 개념과 아이디어를 정리해 드릴게요. 이 힌트를 조합하면 코드를 작성하실 수 있을 겁니다.
이 문제는 Python의 numpy 라이브러리를 얼마나 자유자재로 다룰 수 있는지 확인하는 문제입니다.
1. 핵심 개념: 슬라이싱 (Slicing)과 인덱싱

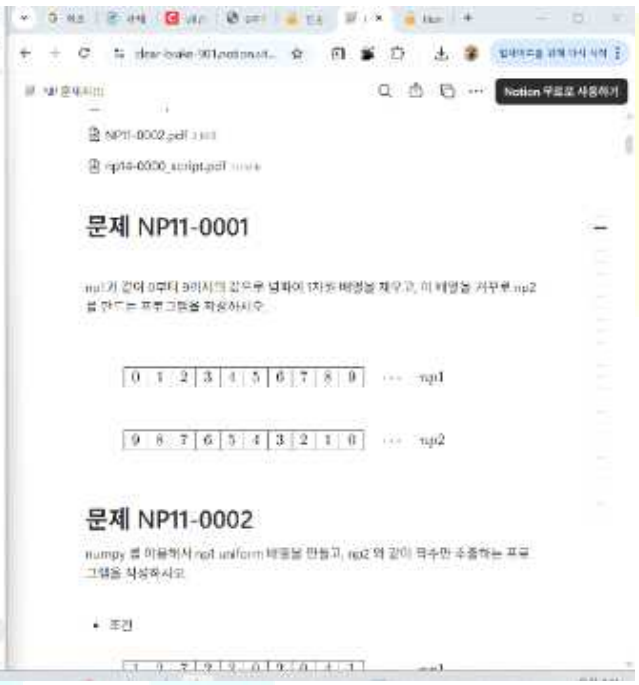

내 풀이에서 논리적으로 틀린 부분이 어디인지 지적해 줘
생각하는 과정 표시
작성하신 코드에서 발생한 오류와 논리적인 문제점을 하나씩 짚어드릴게요. numpy 의 특성과 파이썬의 리스트 문법 차이에서 오는 실수들입니다.
1. 첫 번째 이미지 (np2 생성 시 ValueError)
가장 큰 문제는 for 문의 i 가 무엇을 의미하는지 혼동하신 점입니다.
Python
np2 = [np1[i][::2] if i%2 == 0 else ... for i in np1] # <--- 여기가 문제

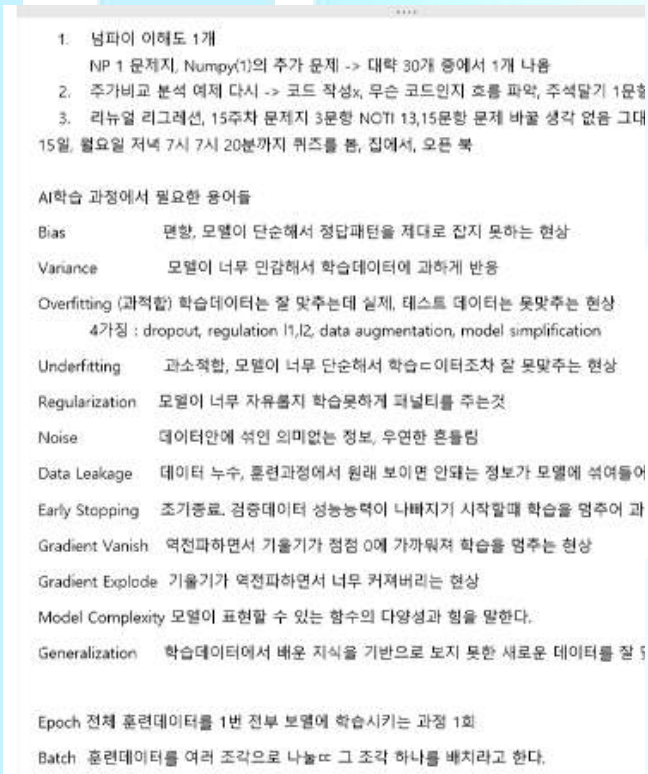


- 정답보다 '이해'중심 학습 강화
  - AI를 정답 제공 도구로 쓰기보다, 풀이에 필요한 핵심 개념 및 공식 힌트를 먼저 받고 스스로 풀이를 완성하는 방식으로 활용하도록 했다.
- 즉각 피드백을 활용한 자기주도 보완
  - 막히는 문제에 대해 시간 및 장소 제약 없이 질문하여 학습 효율을 높이되, 먼저 생각한 뒤 AI를 쓰는 순서를 유지하도록 했다.
- 의존 및 수업 흐름 저해 위험 인지
  - AI에 무조건 의존하면 사고력 약화로 이어질 수 있고, 코드 학습에서는 메소드/라이브러리 이름 혼동등 부작용이 생길수 있음을 인지했다.
- 오프라인 수업 보조 전략 병행
  - 수업 내용은 AI로 구조화 요약하고, 퀴즈로 메타인지 점검까지 연결해 수동 시청/수동 수강에서 벗어나도록 노력하였다.

5주차 활용 AI 서비스: Gemini Pro 3







- 학습 상화에 따른 AI 사용 범위 조절
  - 온라인 강의(강의영상)에서는 AI 사용을 최소화하고, 오프라인 수업에서는 사전학습 키워드 탐색 및 필수 용어 정리, 전체 내용 요약에 제한적으로 사용해 학습 효율을 높이도록 했다.
- 도구 선택 기준 확립
  - 비교적 이해하기 쉬운 설명을 받을 수 있는 ChatGPT 5.2 중심으로 활용했다.

- 부분적 막힘 해소를 통한 문제 해결 속도 향상
  - 문제지를 풀 때 완전히 모르거나, 대체로 풀 수 있으나 일부 단계에서 막히는 구간이 발생하면 ChatGPT 5.2를 활용해 풀이 흐름을 이어가도록 했다.

6주차 활용 AI 서비스: ChatGPT 5.2

구문 (Syntax)

Python

```
numpy.pad(array, pad_width, mode='constant', **kwargs)
```

주요 매개변수

매개변수	설명
`array`	패딩을 적용할 대상이 되는 입력 NumPy 배열입니다.
`pad_width`	각 축(axis)의 앞(before)과 뒤(after)에 추가할 패딩 값의 개수를 지정합니다.
`mode`	패딩을 채우는 방법을 지정합니다. 기본값은 `'constant'` 입니다.
`**kwargs`	선택적 키워드 인수로, 사용하는 `mode` 에 따라 추가로 지정할 수 있는 값들입니다. (예: `'constant'` 모드의 `constant_values`)

```python
import numpy as np
a = np.array([[1, 2], [3, 4]])

# 2차원 배열의 모든 축에 (1, 1) 패딩, 값은 0 (constant_values의 기본값)
b = np.pad(a, pad_width=1, mode='constant', constant_values=0)
# 결과:
# [[0 0 0 0]
#  [0 1 2 0]
#  [0 3 4 0]
#  [0 0 0 0]]
```

X (독립 변수): 연도 (Year)

Y (종속 변수): 기대수명 (Life Expectancy)

연도 (X)	기대수명 (Y)
1930	59
1940	62
1950	70
1960	69
1970	71
1980	74
1990	75
2010	76
2016	78

```python
import numpy as np
from sklearn.linear_model import LinearRegression

# --- 데이터 준비 ---
X_data = np.array([1930, 1940, 1950, 1960, 1970, 1980, 1990, 2010, 2016])
Y_data = np.array([59, 62, 70, 69, 71, 74, 75, 76, 78])

# Scikit-learn 모델 형태로 변환
X = X_data.reshape(-1, 1)
Y = Y_data

# --- 선형 회귀 모델 학습 ---
model = LinearRegression()
model.fit(X, Y)

# --- 1. 직선의 방정식 y = wx + b 출력 ---
w = model.coef_[0]
b = model.intercept_

print("=====================================")
print("1. 선형 회귀 직선의 방정식 (y = wx + b)")
print("=====================================")
print(f"w (기울기): {w:.4f}")
print(f"b (절편): {b:.4f}")
print(f"결과 방정식: y = {w:.4f}x + {b:.4f}")

# --- 2. 1962년 기대수명 예측 ---
year_to_predict = 1962
X_pred = np.array([[year_to_predict]])

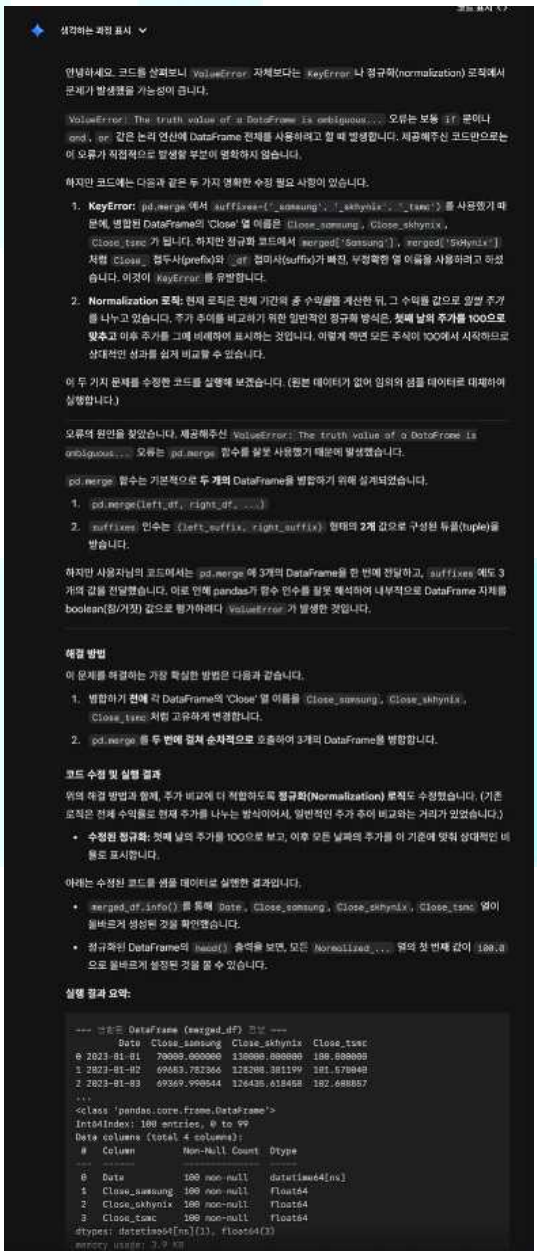
Y_pred = model.predict(X_pred)

print("\n=====================================")
print(f"2. {year_to_predict}년의 기대수명 예측")
print("=====================================")
print(f"예측 기대수명: {Y_pred[0]:.2f}년")
print(f"(참고: 데이터 상 1960년은 69년, 1970년은 71년입니다.)")
```

- 낯선 개념 및 함수의 이해 중심 학습 강화
  - 문제 풀이 과정에서 처음 접하는 함수나 개념이 등장했을 때, AI의 설명 및 예시 제공 기능을 활용해 "사용법"을 넘어 동작 원리와 의미까지 이해하는 것을 목표로 하였다.
- 강의 내용의 공백을 빠르게 메우는 보조 도구로 활용
  - 강의 영상 및 오프라인 수업에서 이해가 불완전했던 부분을 AI에 질문하여, 수업에서 다룬 내용을 추가 설명 및 시각화, 구현 관점으로 보완하였다.
- 단순 복사-붙여넣기 방지(개념 + 구현 결합)
  - 모델 개념은 본인이 이해하고 있다는 전제하에, 부족한 영역인 "코드 구현" 부분에서 AI도움을 받되, 한 줄씩 자신의 지식과 연결해 읽으며 학습 효과를 높이는 방식으로 사용했다.

7주차 활용 AI 서비스: ChatGPT 4.0





- 문제 해결 중심(전처리 및 병합, 시각화 오류 디버깅)
  - 데이터 분석 과제 수행 중 발생한 오류(그래프 누락, 병합 에러, 인덱스 에러)를 해결하기 위해 AI를 "정답"이 아니라 원인 분석 및 해결 방향 제안 도구로 활용하였다.
- 즉시성 기반의 학습 효율 강화
  - 교수자/조교에게 질문 후 답변을 기다리는 방식 대신, AI를 개인 튜터처럼 활용하여 대기시간 없이 즉각 피드백을 받아 과제 수행속도를 높였다.
- 교차 검증을 전제로 한 비판적 활용
  - AI의 할루시네이션 가능성 때문에 답변을 무조건 신뢰하지 않고, 결과를 확인하며 반복검증(교차 검증)하는 태도를 사용 원칙으로 삼았다.

8주차 활용 AI 서비스: Google Gemini Pro 3

AI 기반 대학 수업 혁신 사례의 공유와 확산을 위한 아카이브

발간 KELI

홈페이지 http://www.keliedu.org