CS109A: 데이터 과학 입문

Lecture 1 & Section 1 종합 노트

Pavlos Protopapas, Kevin Rader, Chris Gumb 외

October 26, 2025

▣ **강의명:** CS109A: 데이터 과학 입문

▣ **주차:** Lecture 01

▣ **교수명:** Pavlos Protopapas, Kevin Rader, Chris Gumb

▣ **목적:** Lecture 01의 핵심 개념 학습

□ 핵심 요약

본 문서는 CS109A '데이터 과학 입문' 과정의 첫 번째 강의와 실습 세션을 통합한 종합 노트입니다. 데이터 과학의 정의, 역사적 발전, 5단계 프로세스 등 핵심 이론을 다룹니다. 또한 과정의 상세한 평가 기준, 특히 중요한 출석 정책을 설명합니다. 마지막으로, 파이썬(Python)을 활용한 웹 스크레이핑(Web Scraping) 기초 실습을 단계별로 상세히 안내하여 이론과 실습을 연결합니다.

Contents

1 과정 개요 (Course Overview)

1.1 CS109A: 데이터 과학의 첫걸음

본 과정(CS109A)은 데이터 과학과 인공지능(AI) 분야의 전문가가 되기 위한 여정의 시작입니다. 최신 모델(예: LLM)을 단순히 사용하는 것을 넘어, 그 근간이 되는 **기초 원리(Fundamentals)**를 탄탄히 다지는 것을 목표로 합니다.

이 과정은 무술을 배울 때 기본자세(예: "Wax on, Wax off")를 반복 숙달하는 것과 같습니다. 때로는 지루하게 느껴질 수 있지만, 이 기초가 없으면 복잡한 모델을 제대로 이해하고 활용할 수 없습니다.

1.2 예상 학습 로드맵

본 과정은 데이터 과학의 전체 흐름을 따르며 다음과 같은 순서로 진행됩니다.

1. **데이터 수집 및 탐색 (Weeks 1-2)**: 웹 스크레이핑, 데이터 정제(Wrangling), 탐색적 데이터 분석(EDA) 및 시각화.
2. **회귀 (Regression) (Weeks 3-5)**: K-최근접 이웃(KNN), 선형 회귀, 다중/다항 회귀, 모델 선택(Cross Validation), 추론, 정규화(Ridge, Lasso).
3. **베이지안 모델링 (Bayesian) (Week 6)**: 베이지안 추론 프레임워크, 베이지안 선형 회귀.
4. **분류 (Classification) (Weeks 7-9)**: KNN 분류, 로지스틱 회귀, 계층적 모델링. (중간고사 포함)
5. **데이터 이슈 (Data Issues) (Week 10)**: 결측치(Missingness), 인과 추론(Causal Inference), 편향성 및 윤리.
6. **트리 기반 모델 (Tree-Based) (Weeks 11-14)**: 의사결정 나무, 배깅(Bagging), 랜덤 포레스트(Random Forest), 부스팅(Boosting).

1.3 과정의 3대 목표

본 과정은 이론, 실습, 그리고 실제 영향력이라는 세 가지 축을 중심으로 구성됩니다.

- **이론과 직관 (Theory/Intuition)**: 통계 분석 및 머신러닝의 핵심 개념을 이해하고, 모델 평가 지표를 학습하며, 분석 결과로부터 통찰력을 추출합니다.
- **실습 (Practice)**: 파이썬 라이브러리(Pandas, Scikit-learn 등)를 사용하여 머신러닝 및 딥러닝 모델을 구현하고, 다양한 종류의 데이터를 다루는 법을 배웁니다.
- **영향력 (Impact)**: 데이터 과학을 사용해 실제 문제를 해결하고, 그 과정에서 발생할 수 있는 사회적, 윤리적 영향을 평가합니다.

2 데이터 과학(Data Science)이란 무엇인가?

2.1 데이터 과학의 정의: 3단계 접근

데이터 과학을 이해하는 가장 좋은 방법은 그 역사적 맥락과 구성 요소를 살펴보는 것입니다.

- **1단계 (핵심 요약)**: 데이터 과학은 **데이터로부터 의미 있는 통찰과 가치를 추출**하는 모든 과정을 다루는 융합 학문입니다.
- **2단계 (비유)**: 데이터 과학자는 데이터라는 원석을 캐내어(수집), 불순물을 제거하고(정제), 세공하여(모델링), 아름다운 보석(통찰)으로 만들어내는 장인과 같습니다.
- **3단계 (기술적 설명)**: 정형/비정형 데이터를 수집, 관리, 탐색하고, 통계 및 머신러닝 모델을 적용하여 패턴을 발견하거나 미래를 예측하며, 그 결과를 시각화하여 설득력 있게 전달하는 전 과정을 포함합니다.

2.2 데이터 과학의 역사적 발전 4단계

인류가 세상을 이해하는 방식은 다음과 같이 발전해왔으며, 데이터 과학은 가장 최신의 패러다임입니다.

1. **경험적 관찰 (Empirical Observation) (고대)**
   - 밤하늘의 별을 세거나 농작물 수확량을 기록하는 등, 직접적인 관찰과 경험을 통해 데이터를 수집했습니다.
   - 이는 통계학의 초기 형태라 볼 수 있습니다.
2. **방정식 (Equations) (근대 과학)**
   - 뉴턴, 아인슈타인 등이 등장하며 세상이 작동하는 근본 원리(First Principles)를 수학 방정식으로 설명하기 시작했습니다.
   - 예: $F = ma$, $E = mc^2$
3. **컴퓨팅 (Computation) (20세기)**
   - 1단계의 방정식들이 너무 복잡하여 손으로 풀기 어려워지자, 컴퓨터를 사용하여 시뮬레이션하고 해를 구하기 시작했습니다.
4. **데이터 과학 (Data Science) (현대)**
   - **2단계(방정식)를 건너뛰는 경향**이 나타납니다.
   - 세상의 근본 원리(방정식)를 완벽히 이해하지 못하더라도, **방대한 데이터(1단계)와 강력한 컴퓨팅(3단계)**을 결합하여 세상의 작동 방식을 근사(approximate)하거나 예측합니다.

2.3 데이터 과학의 3대 구성 요소

데이터 과학은 세 가지 핵심 분야가 교차하는 지점에 있습니다.

- **컴퓨터 과학 / IT (Computer Science / IT)**: 데이터 수집, 저장, 처리, 소프트웨어 개발.
- **수학 / 통계 (Math & Statistics)**: 모델링, 가설 검증, 예측의 수학적 기반.
- **도메인 지식 / 비즈니스 (Domain Knowledge)**: 해당 분야(예: 천문학, 의학, 금융)에 대한

전문 지식.

주의사항

도메인 지식의 중요성
도메인 지식이 없는 데이터 과학은 "엉뚱한 문제"를 풀 위험이 큽니다.
한 천문학자가 컴퓨터 과학자에게 데이터 분석을 의뢰했습니다. 한 달 후, 컴퓨터 과학자는 "문제를 완벽히 해결했다"고 했지만, 알고 보니 그는 데이터의 의미와 천문학적 맥락을 전혀 이해하지 못해 완전히 잘못된 문제를 푼 것이었습니다.
데이터 과학은 자동차 정비소에 차를 맡기듯 문제를 던지고 끝내는 것이 아닙니다. **반드시 해당 분야의 전문가와 긴밀히 소통하며 문제 자체를 함께 정의해야 합니다.**

2.4 데이터 과학의 잠재력과 위험성

데이터 과학은 강력한 도구이며, 그에 따른 잠재력과 위험성을 동시에 가집니다.

데이터 과학의 잠재력

- **질병 진단**: 혈액 도말 샘플 이미지로 말라리아 감염 여부 진단.
- **신약 개발**: 언어 모델(LM)을 활용하여 새로운 약물 조합 발견.
- **생성형 AI (Generative AI)**: 텍스트 프롬프트(예: "안경 쓴 그리스인 교수")로부터 이미지 생성.
- **자율 주행**: 야간에도 안전하게 운행하는 자율주행 트럭.

데이터 과학의 위험성과 윤리

- **성별 편향 (Gender Bias)**: 특정 직군(예: 엔지니어) 채용 모델이 남성 지원자에게 유리하게 작동.
- **인종 편향 (Racial Bias)**: 미국 법원에서 사용되는 재범 위험도 예측 모델이 유색인종에게 불리하게 편향됨.

이러한 위험성 때문에 우리는 데이터 과학을 맹목적으로 사용해서는 안 되며, 항상 **비판적 사고 (Critical Thinking)**를 견지하고 모델의 공정성과 윤리성을 점검해야 합니다.

3 데이터 과학의 5단계 프로세스

데이터 과학 프로젝트는 일반적으로 다음 5단계를 순환하며 진행됩니다.

1. **흥미로운 질문하기 (Ask an interesting question)**
   - 가장 중요하고 첫 번째 단계입니다. "데이터가 있으니 뭔가 찾아봐"가 아니라, 명확한 가설이나 과학적 목표를 설정해야 합니다.
   - (예: "우리가 예측/추정하려는 것은 무엇인가?", "모든 데이터가 있다면 무엇을 할 것인가?")
2. **데이터 획득하기 (Get the Data)**
   - 질문에 답하기 위해 필요한 데이터를 수집합니다.
   - (예: "데이터는 어떻게 샘플링되었는가?", "어떤 데이터가 관련 있는가?", "라이선스나 개인정보 보호 문제는 없는가?")
3. **데이터 탐색하기 (Explore the Data - EDA)**
   - 데이터를 시각화하고 요약하며 패턴을 찾습니다. 이 단계에서 많은 시간을 절약할 수 있습니다. (때로는 이 단계만으로도 충분한 답을 얻기도 합니다.)
   - (예: "데이터를 플롯팅해 보았는가?", "이상치(anomaly)나 심각한 오류는 없는가?", "어떤 패턴이 보이는가?")
4. **데이터 모델링하기 (Model the Data)**
   - 데이터의 패턴을 학습하거나 미래를 예측하는 통계/머신러닝 모델을 구축합니다.
   - (예: "모델 구축 (Build) -> 모델 학습 (Fit) -> 모델 검증 (Validate)")
5. **결과 전달/시각화하기 (Communicate/Visualize the Results)**
   - 분석 결과를 비전문가도 이해할 수 있도록 스토리텔링과 시각화를 통해 전달합니다.
   - (예: "우리는 무엇을 배웠는가?", "결과가 말이 되는가(make sense)?", "효과적으로 스토리를 전달할 수 있는가?")

4 CS109A 과정 상세 안내

4.1 교수진 및 조교(TAs)

- **Pavlos Protopapas**: 데이터 과학 석사 과정의 Scientific Director. 천문학과 머신러닝 연구를 수행하며, 요리 자격증을 보유하고 있습니다.
- **Kevin Rader**: 통계학과 선임 지도교수(Senior Preceptor). 학부 교육을 담당하며, 스포츠 및 의학 분야 데이터 분석에 관심이 많습니다. (필라델피아 이글스 팬 - "Go Birds!")
- **Chris Gumb**: 지도교수(Preceptor). 약 30명에 달하는 TF 팀을 조율하고 과정 운영을 지원합니다.
- **Teaching Fellows (TFs)**: 약 30명의 TF가 섹션(실습)과 오피스 아워를 담당합니다.

4.2 학습 철학: "Wax on, Wax off"

□ **예제:**

영화 <베스트 키드>에서 미야기 사부는 제자에게 가라데 대신 자동차 왁스 칠(Wax on, Wax off)만 반복시킵니다. 제자는 불평하지만, 이 무의미해 보이는 반복 작업이 실제 대련에서 방어 동작의 완벽한 기초가 되었음을 깨닫습니다.
CS109A도 마찬가지입니다. 데이터 탐색, 모델 학습, 검증 등 **기본적인 절차를 반복 숙달**시키는 과제가 많을 것입니다. 이는 단순히 코드를 'fit()' 시키는 것을 넘어, **모델이 내부에서 어떻게 작동하는지, 왜 그렇게 작동하는지**를 깊이 이해하기 위한 필수 과정입니다.

4.3 학습 도구

본 과정은 두 가지 주요 플랫폼을 사용합니다.

- **Edstem**: 강의 슬라이드, 섹션(실습) 자료, 공지사항, **토론 포럼(Q&A)**이 이루어지는 메인 허브입니다.
- **Canvas**: 강의 비디오 녹화본, 과제 제출, 공식 일정, **성적 확인**에 사용됩니다.

4.4 도움 받는 방법 (How to get help)

문제가 생겼을 때 다음 순서로 도움을 요청하세요.

1. **Edstem (토론 포럼)**: 가장 빠른 방법. 동료 학생이나 TF가 답변해줍니다. (개인적인 내용 제외)
2. **오피스 아워 (Office Hours)**: 개념 이해나 과제에 대한 심층적인 도움이 필요할 때 가장 좋은 방법입니다.
3. **과정 헬프라인 (Email)**: 수강 변경 등 개인적인 행정 문의.
4. **교수진 (Email)**: 매우 사적인 문제나 민감한 사안.

5 평가 및 주요 정책

5.1 5대 평가 요소 및 비중

학생 평가는 5가지 요소를 합산하여 이루어집니다.

Table 1: *CS109A* 평가 요소 및 비중

평가 요소	비중	세부 내용
숙제 (Homework)	30%	HW0 (1%) + HW 1-5 (29%). 2인 1조(Pair) 작업 권장.
섹션 퀴즈	10%	섹션(실습) 시간 중 실시하는 30분 분량의 퀴즈 2회.
중간고사 (Midterm)	18%	섹션 시간 중 치르는 개념 파트 + 별도의 코딩 파트(Take-home)로 구성.
기말고사 (Final Exam)	22%	3시간 동안 지정된 좌석에서 치르는 시험. (개념 + 코딩)
프로젝트 (Project)	20%	3-5인 1조의 그룹 프로젝트. 공개된(public) 데이터를 활용하여 주제 제안 가능.

5.2 숙제(Homework) 상세

- **HW 0**: 과정 시작 시 배포되며, **선수과목(Prerequisites) 충족 여부를 스스로 진단**하기 위한 목적입니다. (성실히 제출 시 1
- **HW 1-5**: 2인 1조(Pair)로 제출하는 것을 적극 권장합니다.
- **제출 기한**: (별도 공지 없는 한) 매주 화요일 오후 10시.

5.3 선수과목 (Prerequisites) 진단

HW0는 다음 3가지 영역에 대한 준비 상태를 점검합니다.

- **파이썬 (Python) 코딩**:
  - **(필수)** 프로그래밍 경험이 **전혀 없다면** 이 과정을 수강하기 매우 어렵습니다.
  - **(괜찮음)** 파이썬에 능숙하지 않더라도, 다른 언어(예: CS50 수강) 경험이 있다면 따라올 수 있습니다.
- **기초 수학 (Calculus)**: 기본적인 미적분 지식이 필요합니다. (예: Math 1B 수준)
- **기초 통계/확률 (Stats/Probability)**:
  - Stat 104 (데이터 분석 중심) 수강생이 가장 이상적입니다.
  - Stat 110 (수학적 확률론 중심)도 좋지만, 실제 데이터를 다루는 부분은 본 과정에서 새로 배워야 할 수 있습니다.
- **결론**: 수학/통계 지식의 공백은 TF와 교수진의 도움으로 메울 수 있지만, **코딩 경험의 부재**는 심각한 장애물이 될 수 있습니다.

5.4 출석 및 지각 정책

주의사항

CS109A 출석 정책은 매우 중요하며 성적에 직접적인 영향을 미칩니다.

- **출석은 필수입니다 (On-campus 학생)**: 모든 강의와 섹션은 출석이 요구됩니다.
- **성적 등급 자격 (Qualification)**: 출석률은 받을 수 있는 **최고 성적을 제한**하는 "자격" 요건입니다. (이 출석률을 만족한다고 해당 성적을 보장하는 것은 아닙니다.)
  - **A** 등급을 받으려면 → 최소 **66%** (2/3) 출석 필요
  - **A-** 등급을 받으려면 → 최소 **50%** (1/2) 출석 필요
  - **B+** 등급을 받으려면 → 최소 **33%** (1/3) 출석 필요
- **지각 제출권 (Late Days) 획득**:
- 출석(강의 또는 섹션) **4회당 1개의 지각 제출권(Late Day)**을 획득합니다.
- (예: 24회 출석 시 6개의 Late Day 획득)
- **DCE 학생**은 출석 확인이 어려운 점을 감안하여 자동으로 **4개의 Late Day**가 부여됩니다.
- **Late Day 사용**:
- 획득한 Late Day는 숙제(HW) 제출 시 사용할 수 있습니다.
- 한 숙제당 최대 **2개**의 Late Day만 사용할 수 있습니다.

6 실습 (Section 1): 웹 스크레이핑 입문

6.1 웹 스크레이핑이란?

웹 스크레이핑(Web Scraping)은 웹사이트에서 프로그래밍 방식을 통해 자동으로 데이터를 추출하고 수집하는 기술입니다.

첫 번째 실습과 숙제(HW1)는 이 기술을 사용하여 노벨상(Nobel Prize) 웹사이트에서 데이터를 수집하는 것을 목표로 합니다.

6.2 실습 라이브러리

- **requests**: 웹사이트에 접속하여 원본 HTML 코드를 가져오는 라이브러리. (HTTP 요청)
- **BeautifulSoup**: 가져온 HTML 코드를 파이썬이 다루기 쉬운 객체 구조로 변환(Parsing)하고, 원하는 정보를 쉽게 찾도록 도와주는 라이브러리.
- **pandas**: 추출한 데이터를 표(DataFrame) 형태로 정리하고 분석하는 라이브러리.
- **matplotlib**: 데이터를 시각화하는 라이브러리.

6.3 1단계: 웹 스크레이핑 윤리 및 규칙 확인

주의사항

모든 웹사이트가 데이터 수집을 허용하는 것은 아닙니다.

- **robots.txt 확인**: 웹사이트 도메인 뒤에 `/robots.txt`를 붙여(예: `google.com/robots.txt`) 어떤 페이지의 수집을 허용/금지하는지 확인해야 합니다.
- **과도한 요청 금지 (Rate Limit)**: 서버에 부담을 주지 않도록 짧은 시간에 너무 많은 요청을 보내지 않아야 합니다. (예: "1분에 500회 요청 제한")

6.4 2단계: HTML 기초와 브라우저 '검사' 도구

웹사이트는 HTML(HyperText Markup Language)이라는 언어로 구성됩니다.

- **요소 (Element)**: `<tag>`로 시작하여 `</tag>`로 끝나는 전체 구조.
- **태그 (Tag)**: 요소의 종류를 정의합니다. (예: `<h1>`(제목), `<p>`(문단), `<a>`(링크), `<div>`(구역))
- **속성 (Attribute)**: 태그에 추가 정보를 제공합니다. (예: `<a href="..." >`(링크 주소), `<div class="..." >`(요소의 별명))

□ **예제:**

브라우저 '검사' 도구 활용하기

웹사이트에서 원하는 정보(예: 수상자 이름)가 어떤 태그와 클래스(class)로 구성되어 있는지 확인하는 가장 쉬운 방법입니다.

1. 웹페이지에서 원하는 부분에 마우스 오른쪽 클릭

2. **'검사(Inspect)'** 메뉴 선택
3. 개발자 도구가 열리면, 왼쪽 상단의 **'선택 도구(Picker Tool)'** 아이콘(화살표 모양)을 클릭
4. 페이지에서 원하는 요소를 클릭하면, 해당 요소의 HTML 코드가 하이라이트됩니다.

6.5 3단계: `requests`로 데이터 가져오기

먼저 웹페이지의 HTML 소스 코드를 가져와야 합니다.

```
import requests

# 1. 수집할웹사이트  URL
url = "https://www.nobelprize.org/all-nobel-prizes/"

# 2. HTTP GET 요청보내기
response = requests.get(url)

# 3. 상태코드확인이면   (200 성공)
print(f"상태 코드: {response.status_code}")

# 4. HTML 텍스트내용확인일부만   ()
html_text = response.text
print(html_text[:200])
```

Listing 1: requests를 사용한 HTML 코드 요청

- **Status Code 200**: 요청이 성공적으로 완료되었음을 의미합니다. (OK)
- **Status Code 404**: 해당 URL을 찾을 수 없음을 의미합니다. (Not Found)

6.6 4단계: `BeautifulSoup`로 HTML 파싱하기

`response.text`는 다루기 힘든 거대한 문자열입니다. 이를 `BeautifulSoup`를 이용해 "수프(soup)" 객체로 만듭니다.

```
from bs4 import BeautifulSoup

# 1. 'html.parser를' 이용해를 html_text soup 객체로변환
soup = BeautifulSoup(html_text, 'html.parser')

# 2. 원하는정보찾기예   (: 페이지제목 )
page_title = soup.title.get_text()
print(f"페이지 제목: {page_title}")

# 3. CSS 선택자(Selector)로 특정요소찾기
# 예(: 클래스가 'card-prize인' 모든 div 요소)
prize_blocks = soup.select('div.card-prize')
print(f"총 수상블록개수  : {len(prize_blocks)}")

```

```
# 4. 첫번째블록에서텍스트만추출공백 ( 제거)
first_block = prize_blocks[0]
block_text = first_block.get_text().strip()
print(block_text)
```

Listing 2: BeautifulSoup로 HTML 파싱하기

6.7 5단계: 데이터 추출 및 구조화 (노벨상 예제)

여러 개의 수상 블록(prize_blocks)을 순회하며 원하는 정보를 추출하여 리스트와 딕셔너리로 저장합니다.

```
import re # 정규표현식 (Regular Expression) 라이브러리
from collections import defaultdict, Counter

# 1. 람다(lambda)를 이용한간단한헬퍼함수정의
get_title = lambda block: block.select_one('h3').get_text().strip()
get_year = lambda block: re.search(r'(\d{4})', block.select_one('h3').
    get_text()).group(1)
get_description = lambda block: block.select_one('blockquote').get_text
    ().strip()

# 2. 데이터를저장할리스트
nobel_data = []

# 3. 모든수상블록을순회 (loop)
for block in prize_blocks:
    try:
        title = get_title(block)
        year = get_year(block)
        description = get_description(block)

        # 4. 딕셔너리형태로저장
        nobel_data.append({
            'title': title,
            'year': int(year),
            'description': description
        })
    except Exception as e:
        print(f"데이터 추출중오류발생 : {e}")

# 5. 예제() 고유한수상분야찾기
unique_titles = set(item['title'] for item in nobel_data)
print(f"고유 수상분야 : {unique_titles}")

# 6. 예제() 경제학상이처음수여된연도찾기
econ_years = [item['year'] for item in nobel_data if 'Economic Sciences'
    in item['title']]
```

```
first_econ_year = min(econ_years)
print(f"경제학상 최초수여연도  : {first_econ_year}")

# 7. 예제() 연도별수상자수집계    (defaultdict 사용)
winners_per_year = defaultdict(int)
for item in nobel_data:
    winners_per_year[item['year']] += 1 # 으로0 자동초기화됨
print(f"년2023 수상자수 : {winners_per_year[2023]}")
```

Listing 3: 반복문을 통한 데이터 추출 및 구조화

6.8 6단계: pandas로 데이터 프레임 변환

스크레이핑한 데이터(딕셔너리 리스트)는 pandas의 DataFrame으로 변환하면 분석하기 매우 용이합니다.

```
import pandas as pd

# 1. 리스트를데이터프레임으로변환
df = pd.DataFrame(nobel_data)

# 2. 데이터프레임상위개   5 확인
print(df.head())

# 3. CSV 파일로저장
df.to_csv('nobel_prizes.csv', index=False)
```

Listing 4: pandas DataFrame으로 변환 및 저장

6.9 (심화) 비동기(Async) 스크레이핑

수백, 수천 개의 페이지를 스크레이핑할 때는 한 번에 하나씩 요청하면 매우 느립니다.

비동기(Asynchronous) 프로그래밍 (예: asyncio, httpx 라이브러리)은 여러 개의 요청을 동시에 처리하여 속도를 높이는 고급 기법입니다.

이는 서버의 "Rate Limit"(요청 제한)을 존중하면서도 효율적으로 데이터를 수집하기 위해 사용됩니다. (예: "1초에 5개씩" 또는 "한 번에 10개씩 묶어서(batch) 요청")

7 자주 묻는 질문 (FAQ)

- **Q: 이 수업을 청강(Audit)할 수 있나요?**
- A: 네, 가능합니다. 다만 청강으로 수강한 경우, 나중에 동일 과목을 학점 이수(for grade)로 다시 수강할 수 없습니다.
- **Q: 비동기(Asynchronous) 방식(녹화본 시청)으로만 수강할 수 있나요?**
- A: **(대학생)** 허용되지 않습니다. **(대학원생)** 권장하지 않습니다. 출석은 이 수업의 매우 중요한 부분이며, 출석률이 33% 미만일 경우 B+ 이상의 성적을 받을 자격이 박탈됩니다.
- **Q: 선수과목이 부족한데 수강할 수 있을까요?**
- A: **HW0**를 풀어보고 판단하세요. 수학/통계 지식은 도움을 받아 채울 수 있지만, **프로그래밍(코딩) 경험이 전혀 없다면** 수강을 다음 학기로 미루는 것을 강력히 권장합니다.
- **Q: 중간고사 기간에 여행 계획이 있습니다. 시험을 일찍 보거나 미룰 수 있나요?**
- A: 아니요. 중간고사(10/22-24주간)와 기말고사(12/11 예정)는 지정된 날짜에만 치러야 합니다. 일정을 미리 확인하세요.
- **Q: 제가 가진 개인 프로젝트 아이디어를 사용해도 되나요?**
- A: 네, 가능합니다. 단, **데이터가 공개(public)되어 있어야 하며, 3-5명의 그룹 프로젝트**로 진행해야 합니다.