# AI in the Classroom: Transforming Teaching, Learning, and Assessment

Gordon Erlebacher
Department of Scientific Computing
2025-03-05

# Outline

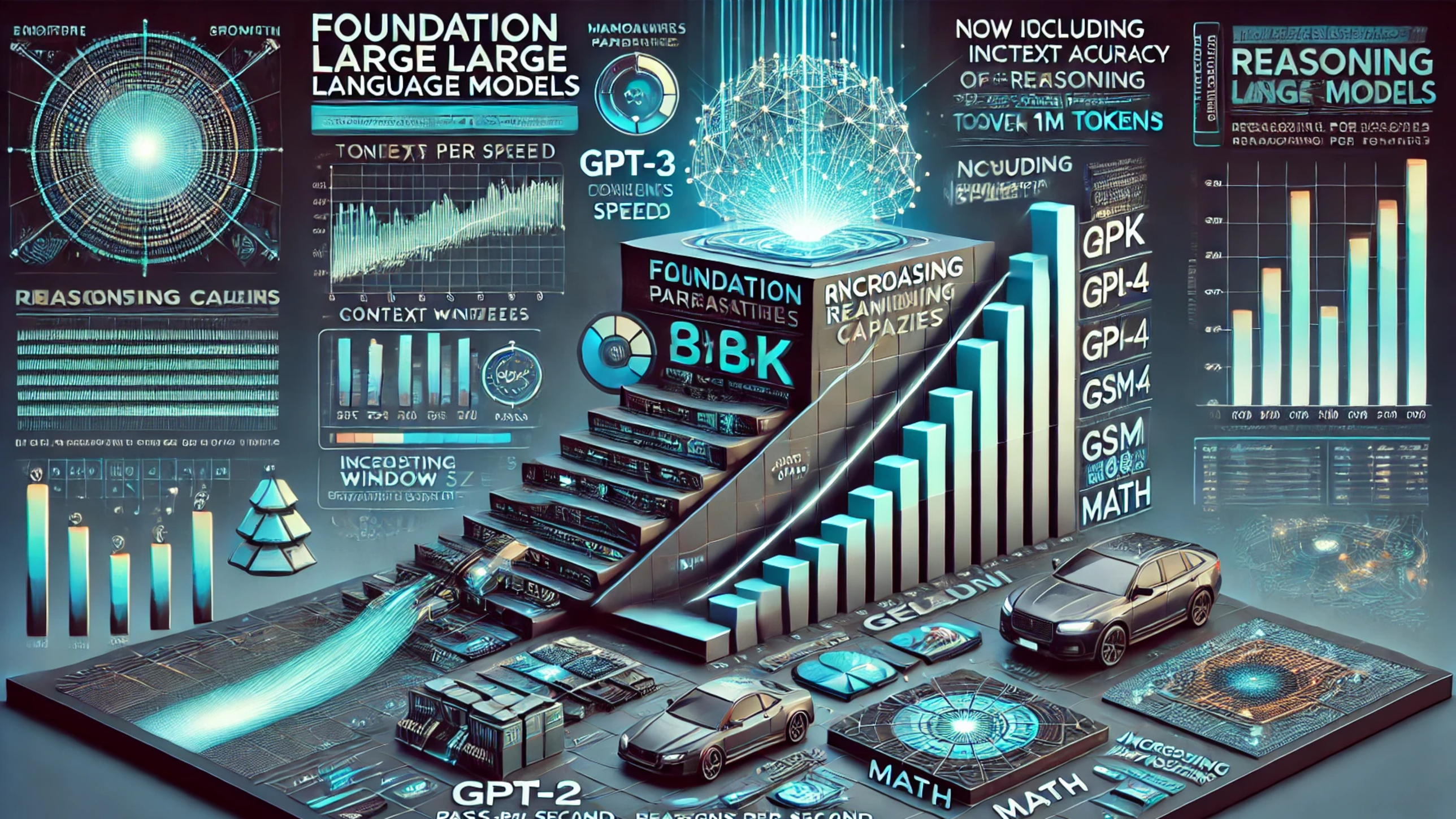
- Recap of Sept 2023 presentations
- Class policy
- Quiz Generation
- Course Materials
- Digital Teaching Assistant
- Automatic Grading and Feedback
- Conclusions

# General Goal

- Illustrate how AI be used to enhance education while maintaining academic rigor?

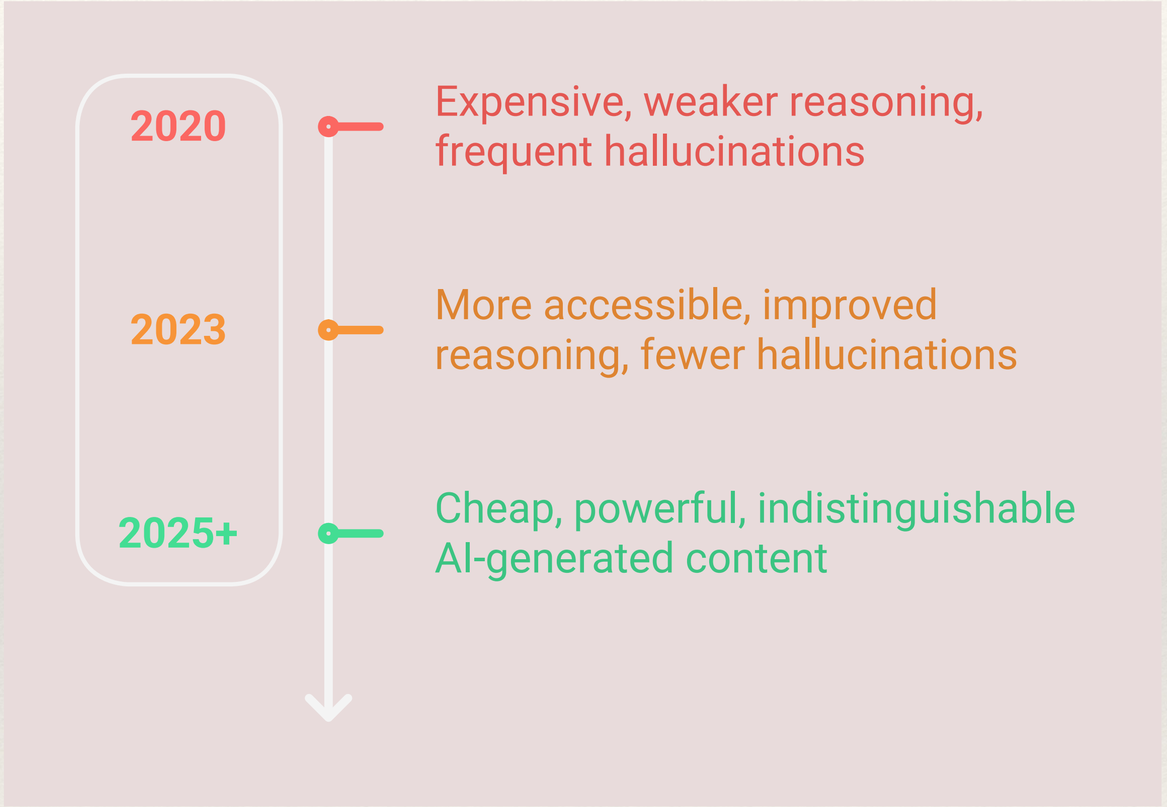
# Bold Predictions (Colloquium 09-06-23)

- AI coding will improve. ✓
- Easier idea-to-code conversion. ✓
- Focus more on algorithms.
- Fewer AI hallucinations. ✓
- Prioritize the "what" over the “how”. ✓
- Manipulate math as easily as natural language.
- Improved presentation software
- Coding language importance will diminish. ✓
- Eliminate the need for a shared oral language.
- Embrace the change!


FOUNDATION
LARGE LARGE
LANGUAGE MODELS
GPT-3
REASONING LANGE MODELS
NOW IOCLUDING
INCTEXT ACURACY
TOOVER 1M TOKENS
NCOUDING
GPK
GPI-4
GPI-4
GSM4
GSM
MATH
FOUNDATION
B'B.K
RNCROASING
CAPAZIES
GELUNI
MATH
MATH
GPT-2

# Develop Policy and Tools Assuming that:

- **LLMs will keep advancing**
  - Cheaper, more powerful
  - better reasoning
- **Hallucinations will persist**
  - Errors will decrease, but not to zero
- **AI Output and Human Work**
  - AI-generated and human content will be indistinguishable

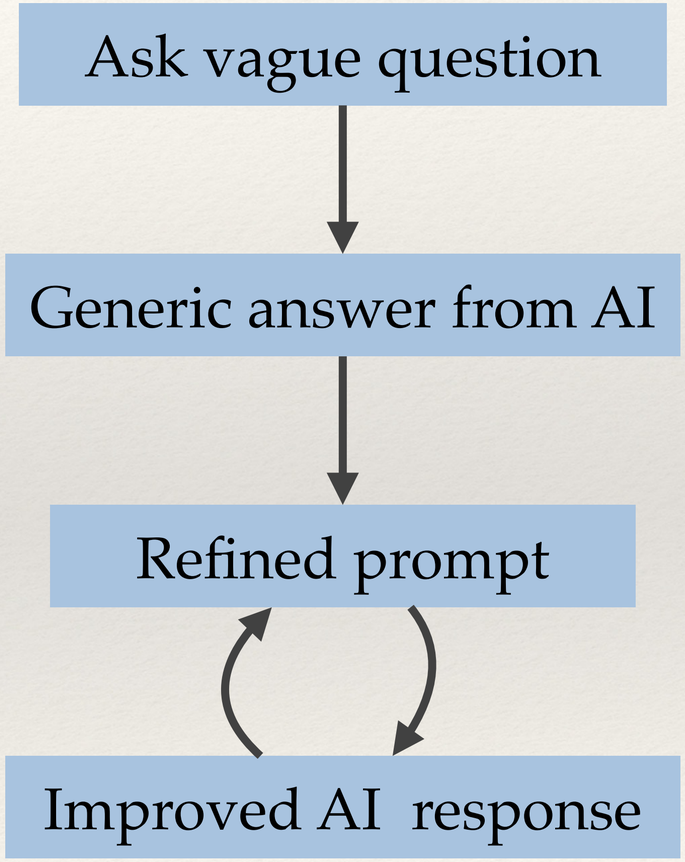


# What Should we Teach our Students?

- **Critical Thinking**
  - Ask, argue, test ideas
- **Validation and Accuracy**
  - Check and validate results
- **AI & Human Balance**
  - Use AI without losing expertise

# AI Usage in Homework: Maximizing Effectiveness

- **Choose the Right AI**
  - Different AIs excel at different tasks (code, explanations, summarization)
- **Adapt to Model Limitation**
  - Free versions require more manual structuring
- **Prompt Effectively**
  - Refining questions improves AI-generated response



# Examples

- An AI generates some results
  - how do you know the result is correct?
- An AI generates a chain of reasoning
  - how do you evaluate?

- I don’t have definitive answers to these questions

# My ToolChest

- **AI Chat Assistants**
  - ChatGPT, Claude, Gemini



- **Research and Citation**
  - Perplexity, NotebookLM



- **Code & Workflow**
  - VSCode + Copilot, Cursor


- **Figure Generation**
  - Dalle-3, Leonardo.ai


# File Formats

- Switched to Markdown
- Easier to write formatted documents
- Default output LLM format
- Use LLM to convert between markdown and other formats
  - tables, latex, another language, another style

# AI Class Policy (Fall 2023)

Artificial intelligence tools are here to stay, and knowledge of their use will grant students a decisive edge in the job market. Therefore, we will encourage using AI tools to learn, experiment, and complete the assignments. Consequently, **the assignments will be more difficult** than in previous years. AI tools enhance your knowledge but **do not replace reading and logical thinking.** You are still required to learn the material, but the **AI can act as a teacher and an assistant.**

AI tools can help with homework. **The students will be limited to specific AIs** (ChatGPT, Claude, Perplexity) to help them with their homework. However, all **conversations must be included in the zipped report.** Failure to do so will lead to point reduction.

# AI Class Policy (Spring 2025)

Artificial intelligence tools are here to stay, and knowledge of their use will grant students a decisive edge in the job market. Therefore, we will encourage using AI tools to learn, experiment, and complete the assignments. Consequently, the assignments will be more difficult than in previous years. AI tools enhance your knowledge but do not replace reading and logical thinking. You are still required to learn the material, but the AI can act as a teacher and an assistant.

**The AI can devise quizzes, and help you solve the problems in the textbook.** AI tools can help with homework. However, all conversations must be included in the zipped report, **or at the very least, a description of which AIs were used and for what purpose has to be stated.** Failure to do so will lead to point reduction.

# Custom LLMs

- All Chatbots have a system prompt
- **Good practice**
  - Provide specialized context to the LLM
- **Avoid repetition**
  - Encode the specialized context into a custom GPT
- **Examples**
  - Detailed Yoga poses
  - Content Generation

**Priming Context**

> You are an expert Python and C++ coder.
Reply in Markdown format.

> Provide examples template usage

# Claude Project (custom prompt): Yoga Master

You are a Yoga Master, with vast experience in breathing, locks, and long poses with and without accessories.

I will provide you with a yoga pose (either in English or Hindi), and you will provide the following:

A description of the pose as a sequence of steps.

A list of the breathing techniques in the pose.

A list and description of the bandhas associated with the pose.

For each step, specify whether to breathe in and out, the associated breathing techniques (pranayama) and bandhas (locks).

# Yoga Master: example

- **Input:** Tree pose
- **Reply:** 🔗

- **Input:** Smart pointers
- **Reply:** 🔗
- The input does not match the intended context ⟶ poor results

# Claude Project (custom prompt): C++Topic

- Project Instructions (🔗)
- Generated via a combination of my input and GPT
- If results are not as expected,
  - use other LLMs to help refine the instructions
- Instructions must be adapted to different LLMs
- Fine-tuning might be necessary for Open-Source models

# C++ Topic Examples (🔗)

- Inheritance
- SFNAE (Substitution Failure is Not An Error)
- Perfect Forwarding
- CRTP (Curiously Recurring Template Pattern)
- Dynamic casting
- STL
- Tree pose (a Yoga position) (poor results)


Instructor
Gordon Erlebacher
Teaching Assistant
Mani Tyagi
Course
Lessons and Homeworks
Homework Demos
Full Course Material
Miscellaneous Topics
ISC5305:
Scientific programming
Canvas
Syllabus
Modules
Grades
Assignments
Source Code
Modern C++ Concepts
Include Headers
Containers and Datastructures
Scientific Computing Examples
Scientific Computing Projects
ISC5305
Scientific Programming

# Content Generation – Key Takeaways

✅ **AI Saves Time on Drafting**

- Generate lesson plans, quizzes, and problems, but refine outputs

✅ **Format Matters**

- Markdown allows easy conversion to Latex, slides, and PDFs

✅ **Adapt Content to Learners**

- Tailor complexity to student needs and learning styles

✅ **Review AI-Generated Content**

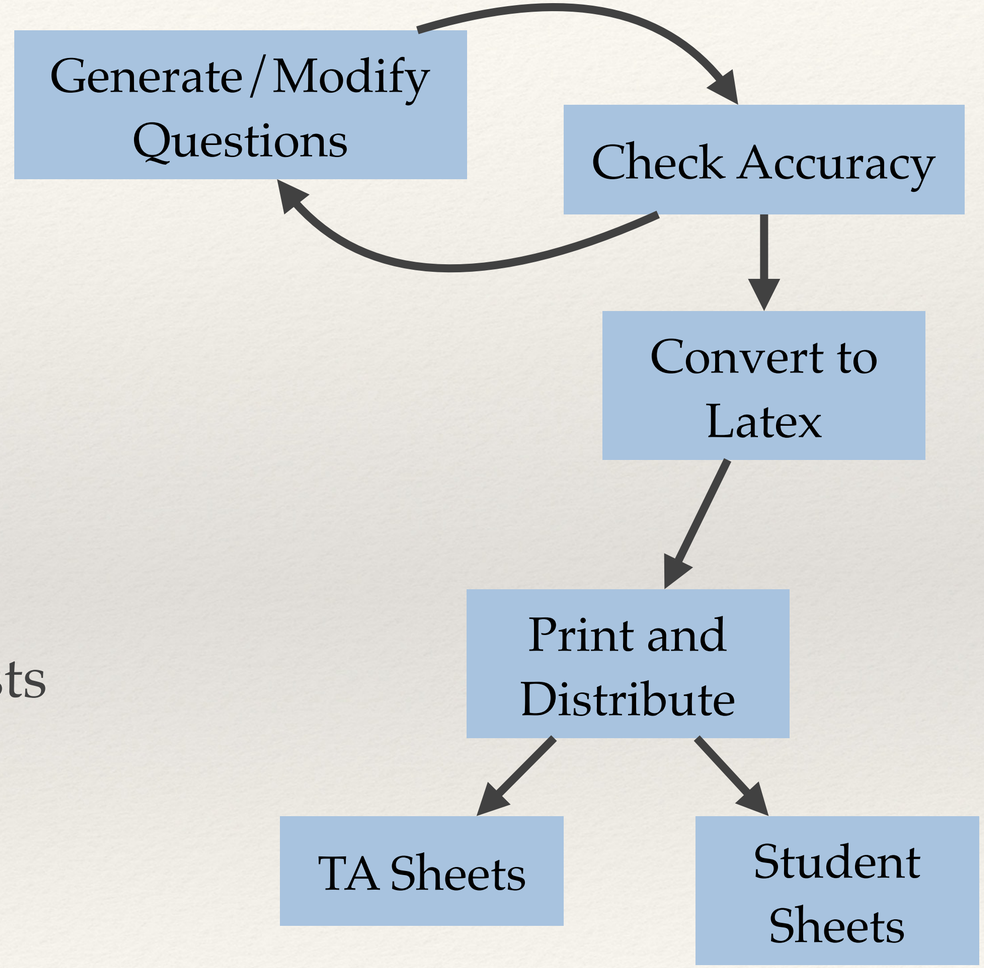
- AI cannot fully replace expertise—always check accuracy

# Quizzes

- Students can use AI for homework assignments
- How to evaluate learning?
- 4-5 quizzes per semester
- Each Quiz: 22 questions
  - Throw away two wrong answers
  - Five choices per question
  - Questions can include code

# Quiz Generation (🔗)

- Custom Artifact or custom GPT
- Uploading homework or chapter
- Modifying individual questions
- Generate TA and student sheets
- **Near Term**
  - Different tests for each student or templatized tests
  - Permute questions
  - Permute options



# Example Question

1. Which of the following best explains why k-means clustering is not suitable for categorical data?

a) K-means uses the Gini index, which is not well-defined for categorical variables.

b) K-means requires decision boundaries, which cannot be formed using categorical attributes.

*c) K-means relies on Euclidean distance, which does not work meaningfully for categorical data.

d) K-means cannot handle missing values, which are common in categorical datasets.

e) K-means always produces high entropy when applied to categorical data.

# Example Question

2. A decision tree is built using the entropy criterion. The root node has 200 samples, with 120 belonging to class A and 80 to class B. What is the entropy at the root node?

a) 1.1

b) 1.02

*c) 0.97

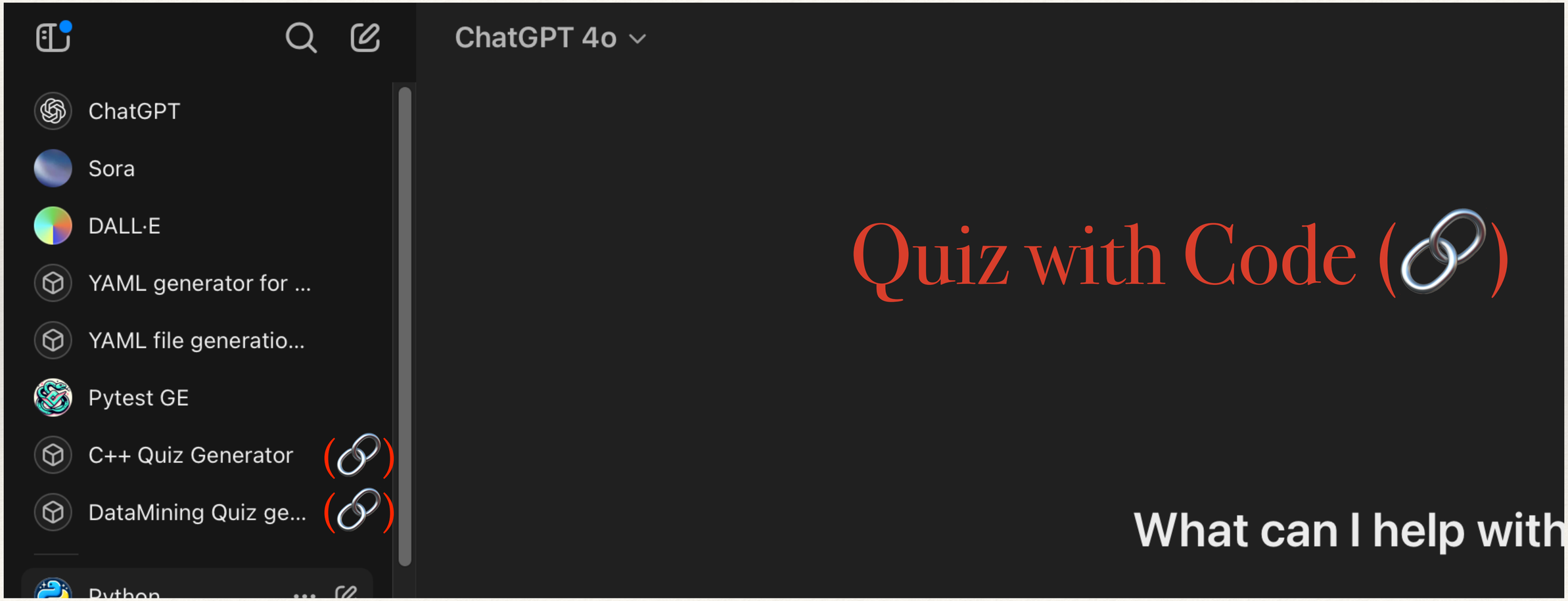
d) 1.22

e) 1.44

| **Permuted Options** |
| --- |
| a) 1.02 |
| b) 1.22 |
| c) 1.1 |
| d) 1.44 |
| *e) 0.97 |

| **Permuted Options** |
| --- |
| a) 1.22 |
| b*) 0.97 |
| c) 1.02 |
| d) 1.1 |
| *e) 1.44 |

# ChatGPT: Custom GPT



# Quiz Structure

| Feature | Student Sheet | TA Sheet |
| --- | --- | --- |
| Questions Format | Standard | Standard |
| Answer Choices | Randomized | Randomized |
| Correct Answers | ❌ Hidden | ✅ Visible |

Assignment batch (b)

First and last name: _____

# Student Sheet

---

**1. Which of the following is NOT a characteristic of agglomerative hierarchical clustering?**

a) It builds a hierarchy of clusters
b) It does not require the number of clusters to be predefined
c) It is guaranteed to produce the same result regardless of linkage type
d) It can use different distance measures
e) It merges clusters step by step

---

**2. In single−link hierarchical clustering, which two clusters are merged?**

a) The two with the highest SSE
b) The two with the lowest SSB
c) The two with the smallest minimum pairwise distance
d) The two with the largest maximum pairwise distance
e) Randomly chosen clusters

Assignment batch (b)

First and last name: _____

# TA Sheet

---

**1. Which of the following is NOT a characteristic of agglomerative hierarchical clustering?**

a) It builds a hierarchy of clusters
b) It does not require the number of clusters to be predefined
*c) It is guaranteed to produce the same result regardless of linkage type
d) It can use different distance measures
e) It merges clusters step by step

Correct answer

---

**2. In single−link hierarchical clustering, which two clusters are merged?**

a) The two with the highest SSE
b) The two with the lowest SSB
*c) The two with the smallest minimum pairwise distance
d) The two with the largest maximum pairwise distance
e) Randomly chosen clusters

Correct answer

# Automatic Quiz Generation – Key Takeaways

✅ **Ensure AI-Generated Quizzes Align with Learning Goals**

- AI can generate diverse questions, but human curation is essential

✅ **Use AI to Generate Variants**

- Shuffle questions and options to prevent memorization-based cheating

✅ **Balance Automation with Oversight**

- Verify accuracy—AI may produce ambiguous or misleading questions

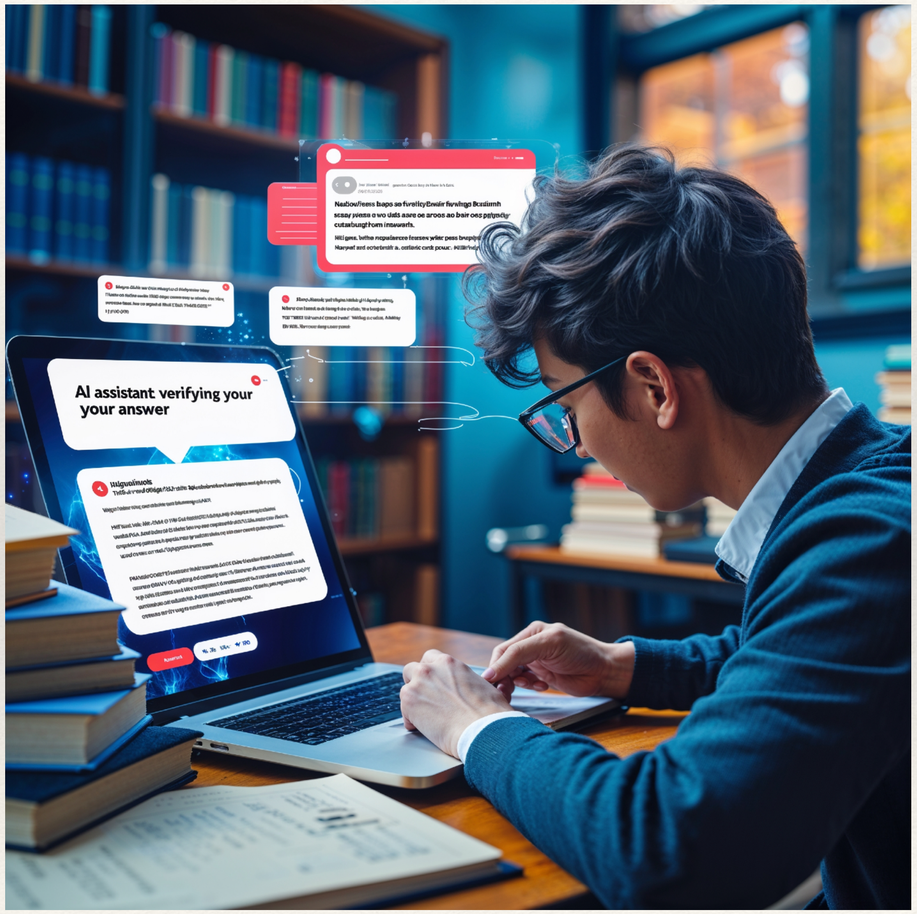
✅ **Leverage Multiple Models**

- Different LLMs generate unique questions—cross-validate for quality

# AI as a Digital Teaching Assistant

- **Enhancing Student Learning with AI**
  - Provides instant feedback and explanations
  - Assists in debugging and code reviews
  - Helps clarify complex concepts through interactive queries
  - Adjustable level of difficulty

- **Students are Encouraged to Use AI to**
  - Explore difficult topics
  - Generate practice quizzes
  - Analyze and break down book exercises
  - Evaluate AI-generated responses



# Chapter Exercises

5. Identify the clusters in Figure 8.36 using the center-, contiguity-, and density-based definitions. Also indicate the number of clusters for each case and give a brief indication of your reasoning. Note that darkness or the number of dots indicates density. If it helps, assume center-based means K-means, contiguity-based means single link, and density-based means DBSCAN.

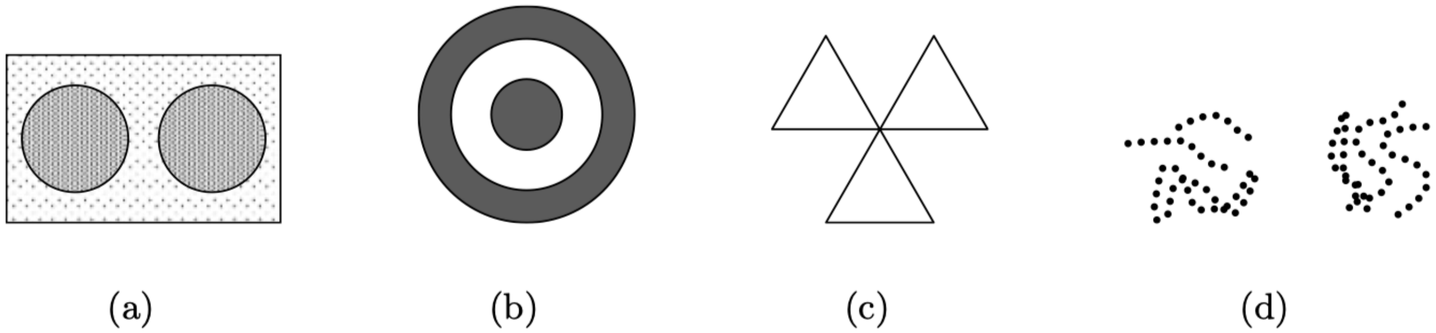



**Figure 8.36.** Clusters for Exercise 5.

Compare AI responses. Identify differences and justify your conclusions:

- ChatGPT (🔗)
- ChatGPT (better context) (🔗)
- GPT (o3-mini-high) (🔗)
- Sonnet 3.7 (🔗)
- Gemini Pro 2.0 (🔗)

Analyze AI outputs, find errors, and refine your understanding.

# Digital Teaching Assistant

- List all the conditional independences in this graph (🔗)
- Can you explain to me how to solve the optimization problem with Lagrange Multipliers? I am confused with the use of inequalities. See image (🔗)
- Answer the questions on the slide (🔗)
- Quiz question (🔗)
- Help me understand this slide (🔗)

# Digital Teaching Assistant – Key Takeaways

## ✅ LLMs Excel at Quick Explanations

- Use AI for concept reviews, debugging code, and explaining problem sets

## ✅ Context Matters

- AI responses improve with well-structured prompts—fine-tune instructions for clarity

## ✅ Limitations Exist

- AI hallucinates and should not replace direct faculty-student engagement

## ✅ Encourage Student AI Use for Learning, Not Answering

- AI should support critical thinking, not replace reasoning

# How to AI-Proof Assignments

- **Question 1:** I teach a course on Data Mining where program assignments are written in Python. How should I structure the programming assignments taking into account that the students are allowed the use of advanced AIs capable of solving a wide range of problems with little effort. Give me range of ideas.
- **Question 2:** Most of the suggested above could be entered into an AI, along with code, to get the answers.

## Replies from

- GPT (🔗)
- Sonnet 3.7 (🔗)
- Gemini 2.0 Flash (🔗)
- Gemini 2.0 Experimental (🔗)

# Grading Python Assignments

- **Required Code Structure**
  - Each question has subquestions
  - Each subquestion is a key in a Python dictionary (hash table)
  - The dictionary is returned to the caller
- **Automatic Testing**
  - Pytest Framework
  - Check answer structure (type) and values

```
def question1():
    answers = {}
    # type: float
    answers['(a) P(x)'] = 3.5
    answers['(b) Q(y | x)'] = 4.6
    return answers

    def question2():
    …
```

# Example Code

- Dictionary-based
- Supports 50+ types
- Example of student code (🔗)

float

Binary Tree

```
def question2():
    answer = {}

    # Answers are floats
    answer["(a) entropy_entire_data"] = 1.42536
    # Infogain
    answer["(b) x < 0.2"] = 0.177393
    answer["(b) x < 0.7"] = 0.355703
    answer["(b) y < 0.6"] = 0.347818
    answer["(b) y < 0.8"] = 0.347818

    # choose one of 'x=0.2', 'x=0.7', or 'x=0.6'
    answer["(c) attribute"] = "x=0.7"  # maximum infogain

    # Use the Binary Tree structure to construct the tree
    # Answer is an instance of BinaryTree
    Lv1 = tree = u.BinaryTree("x < 0.7")

    Lv1L = Lv1.insert_left("y <= 0.6")
    Lv1L.insert_left("B")
    Lv2LR = Lv1L.insert_right("x <= 0.2")
    Lv2LR.insert_right("A")
    Lv3LR = Lv2LR.insert_left("x <= 0.8")
    Lv3LR.insert_left("C")
    Lv3LR.insert_right("B")

    Lv1R = Lv1.insert_right("y <= 0.6")
    Lv2RL = Lv1R.insert_left("y <= 0.3")
    Lv1R.insert_right("A")
    Lv2RL.insert_left("A")
    Lv2RL.insert_right("C")

    answer["(d) full decision tree"] = tree

    return answer
```

# Subset of Handled Types

"dict[int,ndarray]",

“dict[int,list[float]]",

“dict[tuple[int],ndarray]",

"dict[str,set]",

"dict[str,any]",

"dict[int,dict[str,any]]",

"**eval_float**",

"**explain_str**",

"**function**",

"list[int]",

"list[list[float]]",

"list[ndarray]",

"list[str]",

"ndarray",

"scatterplot3d",

**"scatterplot2d",**

"set[ndarray]",

**"DecisionTreeClassifier",**

"RandomForestClassifier",

"StratifiedKFold",

"SVC",

"KFold",

"ShuffleSplit",

"**GridSearchCV**",

# Type: eval_float

- Mathematical expression expressed in Python that evaluates to a float
- **Example**
  - "4 * (a**2 + b**2 + R**2)"
- **Input file**

```
- id:  (b) SSE
  type: eval_float
  tol: 0.01
  locals: {'a': [1., 2.], 'b': [1., 2.], 'R': [1., 2.]}
  answer: "4 * (a**2 + b**2 + R**2)"
```

**Student code**

```
def question3():
    answers = {}
    answers['(b) SSE'] ="4 * (a**2 + b**2 + R**2)"
    return answers
```

# Type: explain_str

- An explanatory string
- **Input file**

```
- id: (c) explain
  type: explain_str
  max_score: 0
  answer: >
      The centroid at 12.5 is farther away from all points than
       any other clusters and will become empty.
```

**Student code**

```
def question3():
    answers = {}
    answers['(b) SSE'] = \
          "SSE generates the best Means clustering."
   return answers
```

# Type: function

- Return a function
- **Input file**

  - id: (c) SSE function
    type: function
    max_score: 20

**Student code**

```
def sum_sq():
    sse = [i^2 for i in range(5)]
    return sum(see)

def question3():
     answers = {}
     answers['(c) SSE function'] = sum_sq
     return answers
```

# Type: DecisionTreeClassifier Instance

- Return a class instance
- **Input File**

  - id: (c) clf
    type: DecisionTreeClassifier

**Student code**

```
from sklearn import DecisionTreeClassifer

def question3():
    clf = DecisionTreeClassifier(max_depth=3, \
          max_features=4, min_samples_leaf=3)
    answers = {}
    answers['(c) SSE function'] = elf
    return answers
```

# Critical Files

- **Input file**
  - hw1.yaml (🔗)
- **Output Pytest files**
  - tests/test_*structure*_preprocessed_hw3_expand.py (🔗)
  - tests/test_*answers*_preprocessed_hw3_expand.py (🔗)
- **Feedback File**
  - results.json

results.json

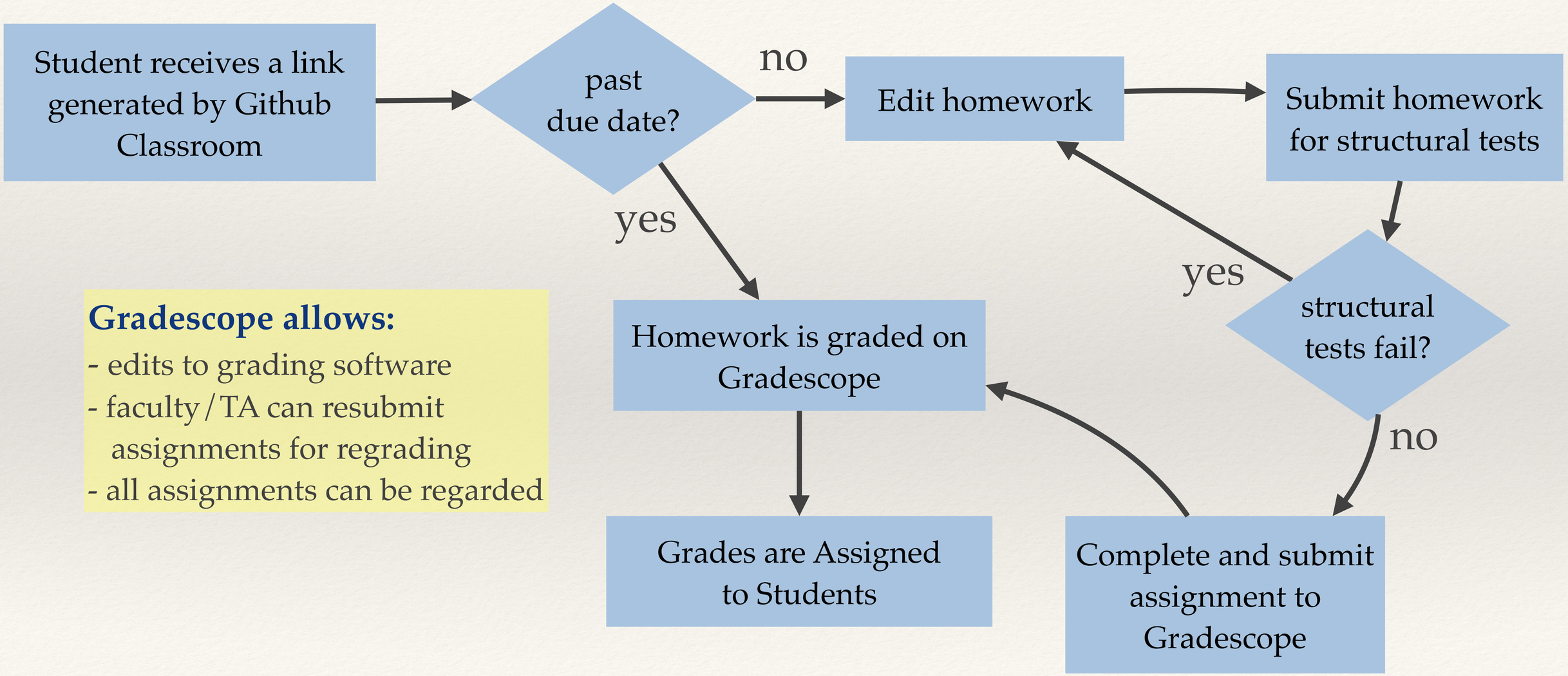

```
question_id :  question1 ,
subquestion_id :  (a) ,
score : 0.0,
partial_score_frac : 0.0,
max_score : 0.0,
answer_type :  str ,
name :  test_structure_question1_a_str ,
output :  Question: 'question1'\nSubquestion: '(a)'\nCorrect
visibility :  visible ,
explanation :  ==Structure tests==:\n- type 'str' is correct
status :  passed


question_id :  question1 ,
subquestion_id :  (a) explain ,
score : 0.0,
partial_score_frac : 0.0,
max_score : 0.0,
answer_type :  explain_str ,
name :  test_structure_question1_a_explain_explain_str ,
output :  Question: 'question1'\nSubquestion: '(a) explain'\
visibility :  visible ,
explanation :  ==Structure tests==:\n- Type 'str' is correct
status :  passed
```

# Strategy with Gradescope

- **Assignment date:** March 1, 2025
- **Due date:** March 15, 2025

- **March 1 - March 14**
  - Submit homework for structure tests (∞)
- **March 15**
  - Grade homework answers and provide feedback

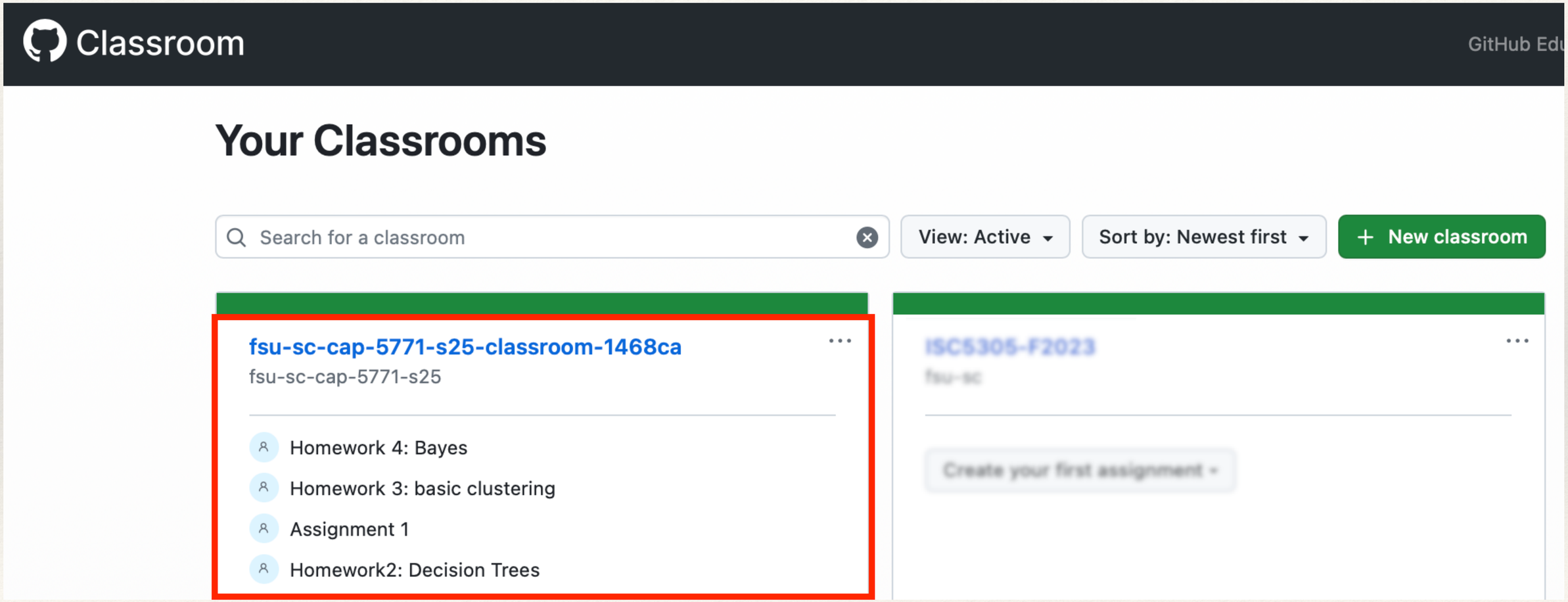
# Homework Assignment Cycle




**Gradescope allows:**

- edits to grading software
- faculty/TA can resubmit assignments for regrading
- all assignments can be regarded

# Github Classroom



# Student Feedback: Correct Answer

**test_answers_question5_a_set_lbrack_str_rbrack (1.72/1.72)**

```
Question: 'question5'
Subquestion: '(a)'
Correct answer type: 'set[str]'
Explanation: ==Structure tests==:
- Type is either 'list' or 'set' (correct).
- All elements are 'str', as required
==Answer tests==:
return_value: student_answer={'group b', 'group a'}
return_value: instructor_answer={'group b', 'group a'}

partial_score_frac: 1.0
max_score: 20
score: 20
rescaled_score: 1.72
```

- Confirm correct structure
- Return instructor and student answer
- Specify score and partial-score information

# Student Feedback: Incorrect Answer

- Provide structural and answer feedback
- Return expected type and type entered by the student
- Student answer was incorrect

**test_answers_question8_a_Matrix_2_str (0/1.72)**

```
Question: 'question8'
Subquestion: '(a) Matrix 2'
Correct answer type: 'str'
Explanation: ==Structure tests==:
- type 'str' is correct
- Answer 'dataset z' is among the valid choices
==Answer tests==:
String element mismatch. Instructor: dataset x, Student: dataset z
return_value: student_answer=Dataset Z
return_value: instructor_answer=Dataset X

partial_score_frac: 0.0
max_score: 20
score: 0.0
rescaled_score: 0.0
```

# Features to use in Future

- Conceptual diagrams of book chapters, articles, blogs (Claude)
- Image input (should be earlier slide)
- Speech input
- NotebookLM, podcast generation for special effects
- Course notes (Claude Artifact for C++)

# Automatic Grading & Feedback – Key Takeaways

## ✅ AI Grading Requires Human Oversight

- Always review AI-graded assignments for fairness

## ✅ Standardize Answer Formats

- Use structured answer formats
  (YAML, multiple-choice, code snippets) for consistency

## ✅ AI Can Provide Instant Feedback

- Automated explanations and grading rubrics help stuedents

## ✅ Ensure Fair & Unbiased Grading

- Maintain fairness,  especially with subjective or open-ended responses

# Embracing AI in Education: Key Takeaways

## ✅ Adapt Teaching Methods

- Focus on problem-solving, reasoning, not memorization

## ✅ Redesign Assessments

- Use *in-class quizzes, project,, and discussions* to test true understanding

## ✅ Leverage AI Wisely

- AI supports grading and content, but verification is key

## ✅ Prepare for the Future

- AI will blend with human work—policies must evolve

# Key Recommendations

## ✅ Use Different AI tools for different tasks

- NotebookLM (🔗), Sonnet 3.7 (🔗), Perplexity (🔗), Cursor (🔗)

## ✅ Do’s & Don’ts of AI in education

- Always check AI output against reality

## ✅ How to AI-proof assignments

- There is no single method. Develop best methods with help from AI
- Assume the students will use AI to the fullest to cut corners

# Some Conclusions

- AI is here to stay
- AI(s) are becoming permanent assistants
- Beware of cognition decline (*)
  - key finding: increase human self-confidence
- Students compete in the workforce against others with field *and* AI knowledge
- Best use of AI is still evolving
- We must train students in best usage of AI tools !!!
- The AI will teach students to ask questions!!!

(*) Leet et al, Generative AI Impact on Critical Thinking in Knowledge Work, https://doi.org/10.1145/3706598.3713778, 2025.