

Guidelines for Projects and Use of AI Assistants

CS 4375: Introduction to Machine Learning
The University of Texas at Dallas
Spring 2026

Overview

In this introductory machine learning course, you will complete four programming projects (subject to change). These projects will give you practical hands-on experience with core machine learning methods and modern reasoning workflows. You will design and implement algorithms, test and debug your solutions, and demonstrate a clear understanding of the concepts.

You are allowed to use AI assistants (large language models and similar tools) as part of your workflow. The purpose is to help you learn how to collaborate with such tools in a careful and professional way. Your own coding ability remains central, and any AI assistance is meant to support your thinking, not replace it.

Why This Matters for Your Future Career

After graduation, you will likely encounter AI assistants in both research labs and companies. Teams already rely on these tools for idea exploration, literature review, code search, debugging, documentation, rapid prototyping, and system design. Strong AI interaction skills will make you more effective as a researcher and as a professional engineer. This course provides a structured setting to practice those skills while you master the underlying concepts.

Expectations in This Course

- You are expected to demonstrate strong coding ability and software engineering practice.
- You should understand every line of code you submit and be able to explain it clearly.
- Use AI assistants for clarification, brainstorming, debugging, and exploring alternate approaches, while keeping ownership of the work.
- Prompting is a basic skill in this course. Coding quality, algorithmic understanding, and clarity take priority.

What You Must Submit

For each project, please submit two items:

1. **Project solution:** Your code and any required write-up as specified in the assignment.
2. **Transcript of your AI interactions:** Include every prompt and every response in chronological order. Do not remove intermediate attempts or mistakes. The full record is important.

Example transcript format:

```
[Prompt 1]: ``How do I implement logistic regression with gradient descent?``  
[Response 1]: ``...``  
[Prompt 2]: ``I am getting a shape mismatch error. What am I doing wrong?``  
[Response 2]: ``...``
```

Optional reflection: Add three to five sentences describing which responses were helpful, which were misleading, and how you decided when to accept or reject suggestions.

Submission Format and File Naming

- Submit the transcript as plain text or PDF. If you export from a chat interface, include the full conversation.
- Name your files with course code, assignment identifier, and your student identifier provided by the instructor. Do not include personal names in file names.
- Place the transcript and your project code in the same submission as separate files.

Anonymity and Privacy

- Do not include your name, email, or other personal identifiers in prompts. Use placeholders such as [STUDENT], [INSTRUCTOR], or [COLLEAGUE] when needed.
- Keep prompts focused on course content. Avoid unrelated personal details.
- Submissions will be kept confidential within the teaching team. Any broader analysis will use anonymized and aggregated data without personal identifiers.

Appropriate Use and Academic Integrity

- You are responsible for the content you submit. Ensure that the code compiles, runs, produces meaningful results and reflects your understanding.
- You may consult public documentation and use AI assistants as learning aids. You must not copy solutions from private repositories or paid solution banks.
- If the tool generates code, you must review it, test it, and modify it as needed. You should be able to explain the logic and design choices.
- You must cite external resources you used, including significant AI-generated code segments or algorithms that informed your implementation.

How We Will Use Your Transcripts

- To verify your workflow and process during grading.
- To identify common pain points and improve instructions, starter code, and examples.
- To study patterns of effective and ineffective AI assistance in a learning setting using anonymized and aggregated summaries.

Tips for Effective AI Interaction

- Ask specific questions that reference your current code or error messages.
- Request explanations and short examples, then implement the approach yourself.
- Validate suggestions with tests. If a suggestion fails, share the failing test and ask why.
- Iterate. Refine prompts as you learn more about the problem.
- Do not share secrets, private data, or credentials in prompts.

Frequently Asked Questions

Is it acceptable to submit code that the model produced?

Yes, but you must review, test, and understand any code you include. Credit the tool when it provided a significant fragment or a key idea. Your understanding and the quality of your implementation are the main evaluation criteria.

Will the transcript affect my grade?

The transcript documents your process. It is not graded for style. It helps us confirm authorship, understand your workflow, and support you more effectively.

What if I forget to record some messages?

Do your best to include the complete record. If something is missing, provide a short note that explains what occurred and continue with a full record thereafter.

Can I use other tools such as linters, formatters, or debuggers?

Yes. You are encouraged to use standard development tools. Document important steps in your workflow when they affect design or results.

Closing Note

This is a machine learning course. You will learn to design and implement algorithms and you will also learn to work productively with AI tools. By documenting your interactions, you strengthen your professional AI interaction skills and help us make the course better for everyone.