DTSC 4525-001: Generative AI and Large Language Models for Data Science

Fall 2026 | 3 Credit Hours | In-Person

Department of Data Science | University of North Texas | Dr. Mohammed Aledhari | mohammed.aledhari@unt.edu

★ Welcome!

Welcome to DTSC 4525! In this course, you'll get hands-on with the technology reshaping data science — large language models, transformers, and generative AI. You'll not only learn how these systems work under the hood, but you'll build, fine-tune, and deploy them yourself — with live demos, in-class coding, and discussions we'll work through together each week. Whether your goal is a career in AI, a research edge, or simply understanding the tools everyone is talking about, this course is designed to get you there. I'm genuinely excited to work with you this semester.

The whole course in one paragraph: Your grade comes from **any 5 of 8** programming assignments (50%) (only these five are graded and receive feedback) plus **one semester project delivered through 4 milestones** (50%) — with **no exams, no quizzes, and no tests**. Around that required core, everything else is optional and built to serve you: **11 fully running micro implementations** **optional — practice only; not graded, no feedback** give you as much hands-on practice as you want, at your own pace, with zero grade pressure and zero deadlines; and **4 extra-credit paths** let you go further — **verbal exams** **run as friendly technical-interview practice** (up to +10%), plus **Conference** (+25%), **Journal** (up to +50%), and **Combined** (up to +60%) publication tracks that turn your class project into a real submission to target venues.

Quick pointers: all assignment topics — the **8 official PAs** and the **11 practice micro implementations** — are listed in the **Programming Assignments** section. For the project, work **individually or in a team of up to 3**; choose from **Dr. Aledhari's suggested ideas** or **your own** — either way, **discuss it with Dr. Aledhari and get his approval before proceeding with any project work**.

In-person requirement: attendance — two free absences, then 5% off your final grade per absence, and **five total absences = F** (full policy on page 28); if you can't reliably attend, consider the **online section (DTSC 4525-401)** — no attendance required.

A living syllabus: I may update this syllabus during the semester if needed (e.g., to improve pacing or incorporate new developments in this fast-moving field). Any changes will always be announced in class and on Canvas, and deadlines will never be moved earlier.

✔ Is This Course Right for You? (Prerequisites)

You're ready for this course if you have:

1. **Math foundations** — comfort with linear algebra, calculus (especially derivatives), probability, and statistics. We'll use these to understand how models actually learn.
2. **Python skills** — you can write and debug Python confidently and are comfortable working in Jupyter notebooks. Basic familiarity with LaTeX (for writing math in your reports) is helpful.

What you do *not* need: No prior coursework in AI, machine learning, deep learning, computer vision, or image processing is required. We'll build everything you need from the ground up — motivated beginners to AI are welcome here.

Not sure whether your background is a good fit? Email me or drop by office hours before the semester starts — I'm happy to help you decide.

➛ How This Course Works (In-Person)

Class meetings	Mondays, 3:30–6:20 PM
Location	NTDP B205, UNT Discovery Park — everything digital lives in **Canvas**
Slides posted	**Mondays by 9:00 PM** each week, so you can preview before class or review after. If there's ever a delay, they'll be up by Tuesday with an announcement.

What this means for you:

- ✓ Real-time explanations, live Q&A, and immediate help when you're stuck.
- ✓ Slides on Canvas by Monday 9:00 PM — preview or review anytime.
- Attendance and participation matter — much of the hands-on learning happens in the room. If you must miss a class, check Canvas and ask a classmate for notes (formal attendance policy in Course Policies). **Tip:** Turn on Canvas notifications so deadlines and announcements never slip by.

A suggested weekly rhythm (flexible — this is just what works well for most students):

- **Before class:** Skim the posted slides/readings so the material isn't brand new.
- **In class:** Engage, code along, ask questions.
- **After class:** Finish the hands-on notebook/lab while it's fresh; complete the week's assignment before its Monday deadline.

Heads up: the rhythm is flexible — the deadlines aren't. Everything is due Mondays at 11:59 PM regardless of our class-day pattern — build your week around that anchor, and the semester stays calm.

✉ Your Instructor

Dr. Mohammed Aledhari *(you're welcome to call me Dr. Aledhari)*

Email	mohammed.aledhari@unt.edu *(best way to reach me outside class)*
Office	Discovery Park, Room E292G
Phone	940-565-3641
Email response time	Within **24–48 hours on business days**

When emailing, please include "DTSC 4525 (In-Person)" in the subject line so I can respond quickly.

A bit about me: *(A short bio and welcome message will be posted in the Week 1 announcement.)*

☛ Office Hours — Come Say Hi!

- **Mondays, 2:00–3:00 PM** — in person at E292G (drop in) or via **Teams/Zoom** *(link posted on Canvas)* by request
- **Or by appointment** — just email me; I'm happy to find a time that works for you, including evenings for working students.

Office hours aren't just for when you're struggling. Come to talk about course topics, AI careers, research opportunities, or anything else. No appointment needed for the regular slot — just drop in.

✉ Teaching Assistant

Gowtham Venkat Eathamokkala — GowthamVenkatEathamokkala@my.unt.edu. Office hours will be posted on Canvas during the first week of class.

Who to contact: any questions or anything else related to the **assignments** must be directed to the **class TA only**. All other matters and questions — including but not limited to **project milestones, grades, extra credits, and course topics and lectures** — must be directed to **Dr. Aledhari only**.

✎ Course Description

This 16-week, in-person course takes you deep into the theory, practice, and societal impact of generative artificial intelligence. You'll start with the mathematical and machine learning foundations that make AI systems understandable, then build upward through deep learning architectures, transformers, and the principles of training and fine-tuning large language models (LLMs). Along the way, you'll get real hands-on experience through programming assignments spanning prompt engineering, retrieval-augmented generation (RAG), multimodal applications, and multi-agent AI systems.

But this course goes beyond the technical. You'll grapple with ethical responsibility, fairness, and bias in AI, and survey cutting-edge research trends and industry applications across text, image, and audio. By combining theory, implementation, and reflection, you'll finish prepared to critically evaluate, design, and deploy generative AI solutions in real-world data science contexts. The semester culminates in a **student project showcase** — live, in-class presentations of your own innovative generative AI application.

✓ What You'll Actually Do

Everything in this course is hands-on, exploratory, and cumulative. Over the semester you will:

- **Build your first LLM app** using an API
- **Implement neural networks and transformer blocks** from the ground up
- **Fine-tune pre-trained models** for domain-specific tasks
- **Experiment with prompt engineering** — and measure hallucination rates yourself
- **Create a RAG pipeline** with a vector database
- **Design multimodal applications** connecting text, images, and speech
- **Develop multi-agent AI systems** for workflow automation and collaboration
- **Finish with a project showcase**, presenting a generative AI application in a domain *you* choose

✤ Why This Course Matters

Generative AI and LLMs are now core skills for data scientists — reshaping healthcare, finance, entertainment, education, and beyond. Employers aren't just asking whether you can *use* these tools; they want people who understand how they work, where they fail, and how to deploy them responsibly. This course bridges theory, practice, and ethics so you leave with genuine technical fluency *and* the critical perspective to use it well.

➤ Learning Outcomes

By the end of this course, you will be able to:

1. **Explain and apply** foundational concepts in linear algebra, probability, statistics, and machine learning that underpin generative AI.
2. **Implement** deep learning architectures (RNNs, CNNs, Transformers) for sequence modeling and classification.
3. **Demonstrate understanding** of transformer architectures and the attention mechanisms behind large language models.
4. **Design, train, and fine-tune** LLMs for domain-specific natural language processing tasks.
5. **Employ prompt engineering strategies** to optimize performance and evaluate hallucination rates in generative AI outputs.
6. **Develop retrieval-augmented generation (RAG) systems** that integrate vector databases with LLMs.
7. **Build generative AI applications** using APIs, integrating multiple modalities (text, image, audio, speech) into unified systems.
8. **Create multi-agent AI systems** demonstrating agent-to-agent communication and workflow automation.

9. **Critically analyze** ethical issues, fairness, and bias in generative AI — and propose mitigation strategies.
10. **Evaluate industry use cases** and articulate emerging trends, preparing for careers and research in generative AI.
11. **Communicate technical results** effectively through written reports, presentations, and applied projects.

How Each Learning Outcome Is Achieved (Alignment Map)

Every learning outcome maps to specific weeks and hands-on work — here's the road map:

Learning Outcome	Aligned Weekly Topics	Key Assignment/Activity
LO1: Apply math/ML foundations	**Weeks 1–2: Math/Stats Refresher + ML Basics**	Foundation for all PAs; optional micro-series practice (microANN, microSVM)
LO2: Implement deep learning architectures	**Weeks 3 & 9: Deep Learning Fundamentals (RNNs/CNNs); GANs & Autoencoders**	**PA1:** Sequence Classification (RNN/CNN)
LO3: Understand transformers & attention	**Week 5: Transformers & Attention Mechanisms**	**PA2:** Implement Transformer Block
LO4: Train & fine-tune LLMs	**Week 6: Training LLMs — Data Preparation & Fine-Tuning**	**PA3:** Fine-Tuning an LLM
LO5: Apply prompt engineering strategies	**Week 7: Prompt Engineering & In-Context Learning**	**PA4:** Prompt Engineering Experiments
LO6: Develop RAG systems	**Week 8: Retrieval-Augmented Generation**	**PA5:** RAG Pipeline
LO7: Build multimodal applications	**Weeks 9–10: Generative Models in Images/Audio; Multimodal Systems**	**PA6:** API-Based Gen AI App · **PA7:** Multimodal API Integration
LO8: Create multi-agent AI systems	**Weeks 11–12: AI Agents, A2A Communication, Workflow Automation**	**PA8:** Multi-Agent AI System
LO9: Analyze ethical issues	**Week 14: Ethical Considerations & Fairness (plus ethics threaded through M2's data-ethics requirements)**	Ethics analysis in project milestones (M2, M3); in-class discussions
LO10: Evaluate industry trends	**Week 13: Advanced Topics — Multimodal Models & Latest Research**	Trend evaluation woven into PA4–PA8 tool choices and your project's related-work section
LO11: Communicate results effectively	**Week 16 (Nov 30): LIVE In-Class Showcase**	**M4:** Final Report + live in-class showcase presentation; PA video walkthroughs all semester

➥ Course Objectives

This course is designed to:

- Provide a comprehensive foundation in the mathematical, statistical, and machine learning principles behind generative AI.
- Cultivate technical proficiency in deep learning, transformer architectures, and training and fine-tuning LLMs.
- Develop applied expertise through structured programming assignments spanning text generation, prompt engineering, RAG, multimodal integration, and agent-based systems.
- Encourage critical engagement with ethical, social, and professional issues — fairness, bias, privacy, and the societal impact of generative models.
- Expose you to state-of-the-art research and industry practice through case studies and real applications.
- Foster creativity and innovation, culminating in an original project that bridges theory and practice.
- Prepare you for career readiness and lifelong learning in a field that reinvents itself every semester.

♥ My Teaching Philosophy: Learn by Doing

You won't just read about generative AI in this course — you'll build it. Every assignment and project exists to give you practice creating and applying real generative AI and LLM techniques. My philosophy is simple: **we learn best by doing.**

What that looks like in practice:

- Every activity is a chance to build something that works.

- **Time commitment:** The standard expectation for a 3-credit course is roughly **9 hours per week total** (for an in-person course: ~3 hours in class + ~6 hours outside — studying, assignments, and project work combined). That outside time is yours to schedule — it works best spread across the week, not crammed into Sunday night.
- **A note on this estimate:** The 9-hour figure is the standard baseline and assumes you enter with the full prerequisite background (math foundations and Python proficiency). Your actual weekly time may be **less or more** depending on your individual background and experience — some weeks may go faster for you, others slower, and that's completely normal.
- Staying on pace with assignments and milestones is the single biggest predictor of success in this course.

Collaboration & Respect

Each of you brings a unique perspective, and our classroom is stronger for it. Together we'll build a course culture based on **open communication, mutual respect, and belonging**. Healthy debate and disagreement — on discussion boards and in feedback — are welcome and encouraged; personal attacks never are. If you ever feel uncomfortable or excluded, please tell me. I'm committed to keeping this course safe, respectful, and welcoming for everyone.

Supporting Your Progress

- I actively monitor how the class is doing and genuinely welcome your suggestions on course activities — this course improves because students speak up.
- Active engagement — asking questions on the discussion board, sharing insights, helping classmates — will deepen your learning (and everyone else's).
- You are not alone in this. I'm here to guide, support, and cheer you on.

My goal is for you to succeed. Reach out anytime you need help, clarity, or just encouragement.

❒ Weekly Schedule & Course Calendar

(Tentative — may be adjusted for your learning needs; changes announced on Canvas, and deadlines never move earlier.)

How the calendar works in this course:

- **Classes meet Mondays, 3:30–6:20 PM; slides post to Canvas by Monday 9:00 PM** each week. First class: week of **Monday, August 17, 2026**.
- **Everything is released on Day 1 (Aug 17):** all 8 programming assignments, all 4 milestone specs, and the project idea list — so you can always work ahead.
- **All deadlines fall on Mondays at 11:59 PM Central**, each with **3 free late days** (no penalty, no questions): assignments get 7 days between deadlines + 3 late days ≈ 10 days of runway each; milestones are spaced three to five weeks apart — the final stretch (M3 → M4) is deliberately the longest — and two deadlines step off Monday: M1 → Tue, Sep 8 (holiday rule) and M3 → Wed, Oct 28 (so it never shares a date with an assignment); every milestone still carries 3 late days.
- **Holiday rule:** If a Monday falls on a university holiday or break, that week has no class meeting that day; slides still post the **following Monday**. This fall that affects two Mondays: **Labor Day (Sep 7)** and **Thanksgiving week (Nov 23)**. No deadlines fall on holidays.

Wk	Monday	Lecture Topics (slides post Mon 9 PM)	Assignment (Released Aug 17 · Due Mon 11:59 PM +3 late days)	Milestone (Released Aug 17 · Due Mon 11:59 PM +3 late days)
1	Aug 17	**Introduction to Generative AI, Course Overview & Syllabus + Math/Stats Refresher (Linear Algebra, Probability & Statistics) — *plus release of ALL assignments, milestones & project ideas***	**PA1–PA8 all released today (Aug 17)**	**M1–M4 specs + project idea list all released today (Aug 17)**
2	Aug 24	**Basics of Machine Learning + Intro to LLMs**	—	—
3	Aug 31	**Deep Learning Fundamentals (RNNs & CNNs)**	—	—
4	Sep 7	**Labor Day — no lecture (posts next Monday)**	—	**M1 (5%): Proposal & Design** Released: Aug 17 · **Due: Tue, Sep 8** · Late window ends: Fri, Sep 11

Wk	Monday	Lecture Topics (slides post Mon 9 PM)	Assignment (Released Aug 17 · Due Mon 11:59 PM +3 late days)	Milestone (Released Aug 17 · Due Mon 11:59 PM +3 late days)
5	Sep 14	**Transformers & Attention Mechanisms**	**PA1 (10%): RNN/CNN for Sequence Classification** Released: Aug 17 · **Due: Mon, Sep 14** · Late window ends: Sep 17	—
6	Sep 21	**Training LLMs: Data Preparation & Fine-Tuning**	**PA2 (10%): Implement a Transformer Block** Released: Aug 17 · **Due: Mon, Sep 21** · Late window ends: Sep 24	—
7	Sep 28	**Prompt Engineering & In-Context Learning**	**PA3 (10%): Fine-Tuning an LLM on a Domain-Specific Dataset** Released: Aug 17 · **Due: Mon, Sep 28** · Late window ends: Oct 1	—
8	Oct 5	**Retrieval-Augmented Generation (RAG)**	**PA4 (10%): Prompt Engineering Experiments** Released: Aug 17 · **Due: Mon, Oct 5** · Late window ends: Oct 8	**M2 (10%): Data, Model & Core Paper** Released: Aug 17 · **Due: Mon, Oct 5** · Late window ends: Oct 8
9	Oct 12	**GANs, Autoencoders & Generative Models (Images/Audio)**	**PA5 (10%): Build & Evaluate a RAG System** Released: Aug 17 · **Due: Mon, Oct 12** · Late window ends: Oct 15	—
10	Oct 19	**Multimodal Systems + Advanced NLP & Evaluation Metrics**	**PA6 (10%): API-Based Generative AI Application** Released: Aug 17 · **Due: Mon, Oct 19** · Late window ends: Oct 22	—
11	Oct 26	**AI Agents: A2A Communication & MCP**	**PA7 (10%): Multimodal API Integration** Released: Aug 17 · **Due: Mon, Oct 26** · Late window ends: Oct 29	**M3 (15%): Implementation & Complete Writeup** Released: Aug 17 · **Due: Wed, Oct 28** · Late window ends: Sat, Oct 31
12	Nov 2	**AI Agents: Workflow Automation in Practice**	**PA8 (10%): AI Agents — A2A & Workflow Automation** Released: Aug 17 · **Due: Mon, Nov 2** · Late window ends: Nov 5	—
13	Nov 9	**Advanced Topics in Generative AI (Multimodal Models, Latest Research)**	—	—
14	Nov 16	**Ethical Considerations & Fairness in AI**	—	—
15	Nov 23	**Thanksgiving Break (Nov 23–29) — no lecture (posts next Monday)**	—	—
16	Nov 30	**Student Project Showcase — LIVE in class + course review & wrap-up**	—	**M4 (20%): Final Report & Presentation — ALL components due** Released: Aug 17 · **Due: Mon, Nov 30** · Late window ends: Dec 3
Finals Week	Dec 7–11	**No class meeting — semester complete!**	—	Nothing scheduled — your LIVE showcase happened in class on Nov 30. (M4 late window for written/code components ended Thu, Dec 3.)

Assignment quick reference *(all released Aug 17; complete any 5 of 8):*

PA	What You'll Build	Due	Late Window Ends
PA1	A recurrent or convolutional neural network for sequence classification (e.g., sentiment analysis or time-series) in Python	Mon, Sep 14	Thu, Sep 17
PA2	A simplified transformer attention mechanism (transformer block), tested on sample data	Mon, Sep 21	Thu, Sep 24
PA3	A pre-trained language model (e.g., DistilGPT2) fine-tuned on a small domain-specific dataset, with documented hyperparameters and results	Mon, Sep 28	Thu, Oct 1
PA4	Designed, tested & evaluated prompt templates across multiple NLP tasks — measuring hallucination rates, accuracy, and output diversity	Mon, Oct 5	Thu, Oct 8
PA5	A retrieval-augmented generation pipeline integrating a vector database (Pinecone, FAISS, or Weaviate) with an LLM for domain-specific Q&A	Mon, Oct 12	Thu, Oct 15
PA6	A generative AI application built on a single API (e.g., OpenAI, Anthropic, Google, or Cohere) for text tasks like summarization or a chatbot	Mon, Oct 19	Thu, Oct 22
PA7	A unified multimodal application integrating multiple AI APIs (text + image generation, or text + speech synthesis)	Mon, Oct 26	Thu, Oct 29
PA8	A multi-agent system where AI agents communicate and collaborate via agent-to-agent (A2A) protocols or MCP, optionally using external data or APIs	Mon, Nov 2	Thu, Nov 5

Helpful registrar dates for your planning: Last day to add: Aug 21 · Last day to drop without a W: Aug 28 · **Last day to drop with a W: Nov 6** (by then, most of your grade is already known — all eight assignments (PA1–PA8) and milestones M1–M3 have all come due — your full grade trajectory is knowable before decision day) · Reading Day: Dec 4 · Final grades submitted: Dec 14.

❄ How Your Grade Works — The 10-Second Version

No exams. No quizzes. No tests. Your entire grade comes from building things:

Component	Weight	The Deal
Programming Assignments	**50%**	Complete **any 5 of 8** (10% each) — you choose which
Final Project	**50%**	Semester-long, individual or teams of up to 3
Optional bonus opportunities	up to +%	Publication tracks & verbal exams — never required

This course is built around **flexibility, real-world skill development, and student choice**. Do the required work well and you earn a top grade; the bonus tracks are there if you want to go further.

Programming Assignments — 50% (Choose Your Own 5)

Eight assignments are offered; **you complete any 5** (10% each). Pick the ones that match your interests and career goals:

1. **PA1:** RNN/CNN for Sequence Classification
2. **PA2:** Implement a Transformer Block
3. **PA3:** Fine-Tuning an LLM
4. **PA4:** Prompt Engineering Experiments
5. **PA5:** Retrieval-Augmented Generation (RAG) Pipeline
6. **PA6:** API-Based Generative AI App
7. **PA7:** Multimodal API Integration
8. **PA8:** AI Agents — Agent-to-Agent (A2A) & Workflow Automation

Flexible submission policy — designed around *your* schedule:

- **All assignments are released early in the semester** — work ahead whenever you're ready.
- **Or follow the official deadlines**, one per assignment.
- **Built-in safety net:** Every assignment includes **up to 3 free late days** after its deadline — no penalty, no questions asked.

Full details (submission rules, tagging, and feedback) are in the Programming Assignments section.

Final Project — 50%

- A **semester-long applied project** — work solo or in a team of up to 3.
- **Staged milestones** with clear expectations and guidance along the way, so you're never wondering "am I on track?"

Full details (milestones, word limits, approval, and bonus tracks) are in the Final Project section.

Free Bonus Practice: The "micro" Series (Optional, Ungraded)

Want to see exactly how AI works — with zero grade pressure? You'll get **11 fully working AI implementations, written from scratch in pure Python with zero dependencies**, each with its dataset included and a matching **interactive visual walkthrough** explaining every function, algorithm, and concept in plain language.

Instead of treating AI as a black box, you'll see how every algorithm works — line by line, input to output. Run any of them on your own computer with nothing but Python installed. No setup, no libraries, no cloud accounts.

Implementation	What You'll See Inside
microANN	Neural networks & backpropagation — the foundation of all deep learning
microCNN	Convolutional networks — how machines see images
microRNN	LSTM recurrent networks — how networks process sequences
microSVM	Support Vector Machines — maximum-margin classification

Implementation	What You'll See Inside
microGPT	The architecture behind ChatGPT
microBERT	The architecture behind language understanding
microTransformer	The original encoder–decoder from *"Attention Is All You Need"*
microGAN	Generative adversarial networks — how AI generates images
microRAG	Retrieval-Augmented Generation — how AI accesses external knowledge
microAgent	AI agents that reason, plan, and use tools
microAutomation	Automated data pipelines with anomaly detection

These cover the foundations of machine learning, deep learning, generative AI, and LLMs — all in one place.

Optional and not required: these 11 implementations are for **practice purposes only** — they are **not graded and receive no feedback**. Practice as much or as little as you like.

Optional Extra Credit — Go Further If You Want

Verbal exams (up to +10%), a Conference Publication Track (+25%), and a Journal Publication Track (up to +50%; combined cap +60%) can push your grade beyond the standard 100% — every one entirely optional. Full rules live in the **Bonus Tracks**, **Optional Bonus Verbal Exams**, and **All Bonus Opportunities — One-Page Summary** sections.

Bottom Line

✓ No exams · ✓ Choose your own 5 assignments · ✓ Work ahead anytime · ✓ 3 free late days each · ✓ One big project you're proud of · ✓ Optional paths to a real publication on your CV

❋ Grading Policy

Grade Breakdown at a Glance

Required (100% of your grade):

Category	Description	Weight
Programming Assignments	Any 5 of 8 assignments (10% each)	50%
Final Project	Multi-stage semester project	50%
Total Required		**100%**

Optional Bonus Opportunities (never required):

Opportunity	Weight	Requires
Bonus Verbal Exams	up to +10%	Up to 2 exams, +5% each
Conference Publication Track	+25%	Completed project + Dr. Aledhari's approval
Journal Publication Track	up to +50%	Completed project + approval + weekly meetings
Combined Conference + Journal Track	up to +60%	One project powering both submissions

Note: The Conference (+25%), Journal (up to +50%), and Combined (up to +60%) tracks are alternatives — you pursue at most one. Verbal exams may be added on top of any path.

Your final grade is based on two required components plus optional bonus opportunities. The structure is designed for **flexibility with clear, fair expectations** — everything you need for a top grade is in the required components; bonuses exist for those who want more.

✈ Programming Assignments (50%)

Programming assignments are the hands-on core of this course — where you build real skills and technical intuition with generative AI, LLMs, multimodal systems, RAG, and AI agents.

Availability & Timeline

- **All eight assignments are released by August 17, 2026** — before the semester even gets rolling.
- Start any assignment whenever you like; finish early or follow the official deadlines.
- This lets you front-load assignment work and shift your focus to project milestones later in the semester if you choose.

Deadlines & Late Policy

Each assignment has an official deadline **plus a 3-late-day grace period** — automatic, per assignment, no justification needed.

When you submit	What happens
✓ On or before the deadline	Accepted, graded, feedback provided *(if tagged [FOR GRADING])*
✓ Within 3 late days after	Accepted with **no penalty**, graded, feedback provided *(if tagged [FOR GRADING])*
After the 3 late days	**Not accepted — automatic zero.** No grading, no feedback, no exceptions.

With early release plus late days, every assignment has weeks of runway. The hard cutoff exists so grading stays fair and timely for everyone — please plan around it.

How Many Are Required

- Complete **any five (5)** assignments for grading — each worth 10%, totaling 50%. **Only these five are graded and receive feedback.**
- **Completing more than five does not increase your grade.** Extra assignments are pure practice.

Why do the optional ones? More reps without grade pressure, exposure to additional tools, and technical confidence.

Submission Tags — Required on Every Submission

Every submission must state your intent by pasting **exactly one** of these tags in the submission comments:

Tag	Meaning
[FOR GRADING]	Counts toward your grade; receives feedback
[FOR PRACTICE ONLY]	Practice; not graded; no feedback

⚠**Default rule:** No tag = automatically treated as [FOR GRADING]. **Double-check your tag before submitting** — an untagged practice attempt will be graded and will count as one of your five, so if you want feedback-free practice, you must tag it [FOR PRACTICE ONLY] explicitly.

Which five count: Only the **first five** submissions tagged [FOR GRADING] (or untagged — grading is the default) and submitted within the deadline + late-day window count toward your grade. Any [FOR GRADING] submissions beyond five are treated as practice. *(Strategic tip: this means the choice of which 5 count is made by submission order — so tag [FOR GRADING] only on the assignments you want counted.)*

Format & Collaboration

- All assignments are **individual** submissions.
- Discussing approaches with classmates is allowed and encouraged — **but the code you submit must be your own work.** (See the Academic Integrity and AI Use policies.)

Key Takeaways

✓ All 8 assignments available by Aug 17 · ✓ You pick which 5 count · ✓ 3 free late days each · ✓ Tag every submission · Hard stop after late days

✪ Transparency in Learning & Teaching (TILT) for Major Assignments

(Planned, but may be adjusted — any updates will be announced right away via Canvas and email.)

These TILT breakdowns exist so you always know **why** each assignment matters, **what** to do, and **how** it will be graded — no guessing, no surprises.

⚠**Submission completeness rule (applies to every PA):** A submission missing any **required component** — the narrated explanatory recording (running code + walkthrough + results), runnable source code, or required files — receives a **zero (0)** for that submission. The rubric percentages below grade the *quality* of complete submissions. Complete work first; polish second.

PA1: RNN/CNN for Sequence Classification — *Due **Mon, Sep 14** (late window ends **Thu, Sep 17**)*

Purpose:

- Gain hands-on practice with classic neural architectures.
- Understand the strengths and limitations of RNNs and CNNs for sequence data.

Task:

- Implement an RNN or CNN model for sequence classification (e.g., sentiment analysis, time-series).
- Train on a provided or chosen dataset.
- Submit code, model outputs, and a 1–2 page summary of results.

Grading criteria:

- Correct model implementation (30%)
- Code functionality & reproducibility (30%)
- Quality of analysis & interpretation (20%)
- Clarity of write-up & presentation (20%)

PA2: Implement a Transformer Block — *Due **Mon, Sep 21** (late window ends **Thu, Sep 24**)*

Purpose:

- Demystify transformers by coding the attention mechanism from scratch.
- Connect theory to practice.

Task:

- Implement a simplified transformer attention block.
- Test on sample data (e.g., toy classification or sequence task).
- Submit annotated code and a brief explanation.

Grading criteria:

- Correct implementation of attention mechanism (40%)
- Code runs successfully & produces expected outputs (30%)
- Clear explanation of how it works (20%)
- Code style & documentation (10%)

PA3: Fine-Tuning an LLM — *Due **Mon, Sep 28** (late window ends **Thu, Oct 1**)*

Purpose:

- Learn how to adapt a pre-trained model to a domain-specific task.
- Practice hyperparameter tuning and reporting.

Task:

- Fine-tune a pre-trained LLM (e.g., DistilGPT2) on a small dataset.
- Document the training process, hyperparameters, and results.
- Submit code, logs, and a 2–3 page report.

Grading criteria:

- Correct fine-tuning procedure & reproducibility (40%)
- Documentation of hyperparameters & rationale (20%)
- Evaluation of results & discussion (25%)
- Clarity of report (15%)

PA4: Prompt Engineering Experiments — *Due **Mon, Oct 5** (late window ends **Thu, Oct 8**)*

Purpose:

- Understand how prompt design influences LLM outputs.
- Explore issues like hallucinations, accuracy, and diversity.

Task:

- Design and test multiple prompt templates for NLP tasks (e.g., summarization, Q&A).
- Evaluate and compare results.
- Submit code/notebook and a 2–3 page reflection.

Grading criteria:

- Variety & creativity of prompts tested (30%)
- Systematic evaluation of outputs (30%)
- Insightful reflection on strengths/weaknesses (25%)
- Clarity of documentation (15%)

PA5: Retrieval-Augmented Generation (RAG) Pipeline — *Due **Mon, Oct 12** (late window ends **Thu, Oct 15**)*

Purpose:

- Learn how to combine information retrieval with LLM generation.
- Build systems closer to real-world enterprise applications.

Task:

- Build a RAG pipeline using a vector database (e.g., Pinecone, FAISS, or Weaviate).
- Demonstrate Q&A over a domain-specific dataset.
- Submit code, screenshots/examples, and a short report.

Grading criteria:

- Correct RAG pipeline integration (40%)
- Functionality of retrieval & generation (30%)
- Demonstration with meaningful examples (20%)
- Clear explanation of pipeline (10%)

PA6: API-Based Generative AI App — *Due **Mon, Oct 19** (late window ends **Thu, Oct 22**)*

Purpose:

- Practice integrating external APIs into practical applications.

- Explore the application-development workflow.

Task:

- Develop a generative AI app using an external API (e.g., OpenAI, Gemini, Anthropic, Cohere).
- Example apps: chatbot, summarization tool, content generator — or your own idea.
- Submit code/app and a 2–3 page description of functionality.

Grading criteria:

- Working integration with chosen API (40%)
- Creativity & usability of app (25%)
- Code clarity & documentation (20%)
- Report quality (15%)

PA7: Multimodal API Integration — *Due **Mon, Oct 26** (late window ends **Thu, Oct 29**)*

Purpose:

- Explore how multiple modalities (text, image, speech) can be combined.
- Understand practical challenges of multimodal integration.

Task:

- Create an application that combines two or more APIs (e.g., text+image, text+speech).
- Demonstrate with examples.
- Submit code and a 2–3 page report.

Grading criteria:

- Successful integration of multiple APIs (40%)
- Functionality & creativity of multimodal system (30%)
- Quality of demonstration (20%)
- Report clarity (10%)

PA8: AI Agents — A2A & Workflow Automation — *Due **Mon, Nov 2** (late window ends **Thu, Nov 5**)*

Purpose:

- Learn how to design AI systems where agents collaborate.
- Gain experience with frameworks and protocols like LangChain, A2A, and MCP.

Task:

- Create a multi-agent system where agents communicate and collaborate.
- Optionally integrate external APIs or datasets.
- Submit code, run examples, and a 3–4 page project report.

Grading criteria:

- Correct design of agent architecture (40%)
- Demonstrated functionality of communication/collaboration (30%)
- Depth of analysis & reflection in report (20%)
- Code quality & documentation (10%)

✻ Final Project (50%)

Project ideas — your choice, my approval: choose one of the **suggested project ideas posted on Canvas** or **propose your own**. Either way, send your idea to Dr. Aledhari **as soon as possible — within the first two weeks of the semester (by Mon, Aug 31)** — to get his approval confirming the idea **aligns with the course requirements, is doable, and can be implemented and completed within the semester**.

Scope & Supervision

- Your project must be **developed exclusively for this course** under Dr. Aledhari's direct supervision.
- Projects from other courses, independent studies, theses, dissertations, or external research programs **cannot** count toward or replace any part of this project.
- Every project — standard or bonus-track — must fully satisfy **all** required milestones and deliverables.

Team Structure

- Work **individually or in groups of up to three (3)**.
- Group membership must be declared at project approval time.
- All members share responsibility and normally receive the same grade, unless Dr. Aledhari determines otherwise (e.g., documented unequal contribution).

Choosing Your Project

Pick one path — both require approval:

1. **Propose your own** — any idea aligned with course topics and learning objectives. *(Most students find this the most rewarding path — build something you actually care about.)*
2. **Choose from the instructor's list** — Dr. Aledhari releases suggested project ideas to all students by **August 17, 2026**.

Mandatory Approval — Before Any Work Begins

- Discuss your idea with Dr. Aledhari (**by email or a Teams/Zoom office-hours call**) and obtain formal written approval by email before starting.
- Approval must be secured **within the first two weeks of the semester (by Mon, Aug 31)** — well before Milestone 1 (Proposal & Design).
- Work done before approval is at your own risk and may need revision or discarding.

This isn't bureaucracy — it's how I make sure your project is scoped to succeed and, if you're bonus-track curious, publication-viable from day one.

☎ Bonus Tracks — Declaration & Commitment

Declaring Your Intent

If you plan to use your project for the **Conference (+25%)**, **Journal (up to +50%)**, or **Combined (up to +60%)** track, you must declare this in the **project questionnaire on Canvas**. This lets Dr. Aledhari provide the right guidance, expectations, and requirements for your path.

No declaration = standard course project by default (which is a perfectly great outcome — bonus tracks are for those who want the extra challenge).

Choosing your target venue (Conference and/or Journal tracks): before locking in your project idea, check which conference or journal is your **best target** and make sure your idea and topic fall **within that venue's scope and alignment** — so you don't spend the semester on work that would face immediate rejection at the target venue. **Discuss your target venue with Dr. Aledhari in advance — within the first two weeks of the semester — to get his approval and advice.**

Journal Publication Track — Know What You're Signing Up For (up to +50%)

This path offers exceptional value — a real publication on your CV, developed with substantial dedicated mentorship — but it demands genuine commitment:

- **Weekly meetings with Dr. Aledhari (in person at Discovery Park; Teams/Zoom backup) are required** (up to twice weekly when needed).
- You must show **real, measurable progress every week**.
- Weekly meetings **without meaningful progress** result in immediate cancellation of the track and ineligibility for its extra credit.
- If you can't commit to consistent weekly progress, choose the standard project — there's no penalty for aiming there.

Eligibility Reality Check

Selecting a bonus track does not guarantee extra credit. Credit is awarded only when all requirements, commitments, and expectations for that track are fully met.

Bottom Line

Bonus tracks are exceptional opportunities — but they require clear intent, formal declaration, and sustained weekly effort. The standard path to an A is the required work done well; the bonus paths are for going beyond it.

❖ Project Milestones, Expectations & Length Guidance

Your project unfolds in four staged milestones, so you always know what's next and are never facing a blank page in November. Expectations below apply to all projects; publication-track projects meet the same milestones with added rigor noted inline.

Milestone	What's Expected	Min. Words*	Deadline†	Weight
1. Proposal & Design	Define the problem statement, motivation, related-work context, research questions or system goals, and proposed methodology or architecture. *Publication-track: articulate novelty and evaluation plans.*	≥ 800	**Tue, Sep 8**	5%
2. Data, Model & Core Paper	Describe datasets, preprocessing, and ethical considerations; justify model selection with reproducibility details and baseline comparisons. Then draft the core paper sections: Abstract, Introduction, Related Work, and Development Environment (data sources, dataset sizes, hardware, software, compute), following academic writing and citation standards.	≥ 2,000	**Mon, Oct 5**	10%
3. Implementation & Complete Writeup	Build the system/model and run experiments — clearly describe setup, metrics, and results (publication-track: include ablations and comparisons where appropriate). Deliver a near-complete paper draft including Results, Discussion, Conclusion, and Future Work, emphasizing interpretation of results, limitations, and broader implications.	≥ 6,000	**Wed, Oct 28**	15%
4. Final Report & Presentation	Polished, submission-ready report **plus a live in-class showcase presentation during our final class meeting (Nov 30)**. *Publication-track: follow full academic paper structure and presentation standards.*	≥ **9,000**	**Mon, Nov 30**	20%
Total				**50%**

† Deadline & Late Policy for Milestones: All deadlines are 11:59 PM Central. **Every milestone — including the final report and showcase materials — carries the same 3-late-day grace period as programming assignments:** submissions within three (3) days after the deadline are accepted with no penalty, no justification needed. After the 3 late days, the milestone receives a zero. Note on M4: the long runway between M3 (Oct 28) and M4 (Nov 30) — the largest spacing in the

course — includes Thanksgiving Break as built-in buffer; no deadline touches the break. Optional mid-November check-in (ungraded, encouraged): during the long M3→M4 runway, send me a half-page status note or drop by office hours so your final report benefits from early feedback.

*** Word Count Policy**

- Word counts **exclude** references, tables, figures, and supplementary materials — the minimums refer to your main written text only.
- Listed counts are **minimums**, not targets to stop at. Exceeding them is encouraged whenever it improves clarity, rigor, or completeness.
- **Natural length varies by project:** different topics and target fields (e.g., a systems-focused conference paper vs. a survey-style journal manuscript) naturally produce different final lengths — that's expected and fine. However, **every final project delivery must contain no fewer than 9,000 words** of main text, in addition to (not counting) figures, tables, references, and supplementary materials.
- Submissions below the minimum word count are considered **incomplete**.
- *The counts are cumulative in spirit: each milestone builds on the last, so by Milestones 3–4 you're expanding and refining earlier writing, not starting from scratch.*

▲Critical Rule

Missing the Final Report & Presentation (including its 3-late-day window) results in an automatic zero (0/50) for the entire project, regardless of performance on earlier milestones. The final deliverable is the project — earlier milestones exist to get you there.

✍ Labeling Directions for Submissions & Deliverables

To keep grading fair, organized, and consistent, all submissions (files, reports, code, and zipped folders) must follow the required naming conventions.

Why this matters:

- Correct labels make sure your work is identified quickly and graded without delays.
- ▲Submissions with the wrong file name/label receive a **25% grade reduction** for that assignment — a completely avoidable loss, so please copy-paste the labels below.

Recommended Submission Workflow

1. **Create an empty folder** on your computer.
2. **Name the folder** using the required label (e.g., PA1-DTSC4525-Firstname Lastname or M1-DTSC4525-Firstname Lastname).
3. **Place all required files** (code, report, data, video, etc.) inside.
4. **Zip the folder** and upload to Canvas — don't forget your [FOR GRADING] or [FOR PRACTICE ONLY] tag in the submission comments for programming assignments.

When we unzip your submission, we should immediately see a correctly labeled folder with all your files neatly inside.

Example:

- ✓ PA1-DTSC4525-Alex Smith.zip → unzips to → PA1-DTSC4525-Alex Smith (all required files inside)
- Randomly named zip, loose files, or wrong folder labels

Quick tip: Double-check the file name and contents before uploading — the easiest points you'll ever protect. Copy the exact label and replace the name with yours: **space between first and last name; no hyphens, dashes, or merged names.**

Programming Assignment Labels & Deadlines

Format: PA#-DTSC4525-Firstname Lastname · **Example:** PA3-DTSC4525-Jane Doe

Label	Due (11:59 PM CT)	Late Window Ends
PA1-DTSC4525-Firstname Lastname	Mon, Sep 14	Thu, Sep 17
PA2-DTSC4525-Firstname Lastname	Mon, Sep 21	Thu, Sep 24
PA3-DTSC4525-Firstname Lastname	Mon, Sep 28	Thu, Oct 1
PA4-DTSC4525-Firstname Lastname	Mon, Oct 5	Thu, Oct 8
PA5-DTSC4525-Firstname Lastname	Mon, Oct 12	Thu, Oct 15
PA6-DTSC4525-Firstname Lastname	Mon, Oct 19	Thu, Oct 22
PA7-DTSC4525-Firstname Lastname	Mon, Oct 26	Thu, Oct 29
PA8-DTSC4525-Firstname Lastname	Mon, Nov 2	Thu, Nov 5

(All released Aug 17. Complete any 5 — see Grading Policy.)

Final Project Milestone Labels & Deadlines

Format: M#-DTSC4525-Firstname Lastname — same simple pattern as your programming assignments. *(Write the full milestone title on your report's cover page; the label stays short.)*

Example: M1-DTSC4525-Jane Doe.zip → unzips to → M1-DTSC4525-Jane Doe (all files inside)

Team projects (up to 3 students): Even when working as a team, **each team member must individually submit an identical copy of the project** to Canvas:

- Label your submission with **your own name** (the submitter): e.g., Jane Doe submits M1-DTSC4525-Jane Doe, and teammate John Smith submits the same content as M1-DTSC4525-John Smith.
- **Inside the project files** (report cover page and code header), **all team members' names must be listed**.
- All copies must be **truly identical** — differing copies across teammates will be treated as an academic integrity concern.
- Why this rule? It guarantees every member has the full project on record, grades and feedback post cleanly to each student's own Canvas entry, and no one's grade depends on a teammate remembering to upload.

Label	Milestone (title goes on your cover page)	What You Submit	Due (11:59 PM CT)	Late Window Ends
M1-DTSC4525-Firstname Lastname	Proposal & Design	Proposed idea & initial design	Tue, Sep 8	Fri, Sep 11
M2-DTSC4525-Firstname Lastname	Data, Model & Core Paper	Datasets, ethics & model justification + core paper sections (Abstract, Intro, Related Work, Dev Environment)	Mon, Oct 5	Thu, Oct 8
M3-DTSC4525-Firstname Lastname	Implementation & Complete Writeup	Working code & experiments + near-complete draft (Results, Discussion, Conclusion, Future Work)	Wed, Oct 28	Sat, Oct 31
M4-DTSC4525-Firstname Lastname	Final Report & Presentation	Written technical report + **live showcase presentation (in class Nov 30)**	Mon, Nov 30	Thu, Dec 3

⚠ **Missing the final milestone results in a zero for the entire project** (see Grading Policy).

➨ Programming Assignment Submission Guidelines

To earn full credit, your submission must include all required components. Follow this checklist carefully:

Python Source Code

- Submit .py or .ipynb (notebook) files.

- **Your code must run correctly on the instructor's computer** — include every dataset, library reference, and helper file it needs.
- *Test it the way we'll run it: restart your kernel or open a clean environment, and run start to finish. "It worked on my machine" is the most common — and most preventable — grading problem in programming courses.*

Video Demonstration

Record a short video explaining your work — this is how you show you truly understand your code:

- **Your voice must be included** (no silent demos).
- **Show yourself clicking "Run"** and generating the outputs live.
- **Explain each main section** of the code, the results, and any plots.
- Missing narration, missing "Run," or skipped sections = **no feedback**.

Think of it as teaching me how your program works — it doesn't need to be fancy, just clear and complete. *(Bonus: this is exactly the skill the optional verbal exams and your final presentation build on.)*

Required Files

- Include **all** datasets, model files, and helper scripts needed to run your code.
- If the code can't run because a file is missing, your work can't be reviewed.

No External Links

- **Do not submit links** to Google Drive, Dropbox, GitHub, or other cloud storage — all files (code, video, data) must be uploaded **directly to Canvas**.
- *(Using GitHub for your own version control while developing? Great habit — keep doing it. Just upload the actual files to Canvas when you submit.)*

What a Complete Submission Looks Like

For PA6 submitted by Jane Doe — PA6-DTSC4525-Jane Doe.zip unzips to:

PA6-DTSC4525-Jane Doe/

|— PA6-DTSC4525-Jane Doe.py (or .ipynb)

|— dataset.csv

|— helper_functions.py

|— ...

'— PA6-DTSC4525-Jane Doe.mp4 (video demo)

✓ Pre-Submission Checklist

☐ Code file(s) included (.py/.ipynb) and run start-to-finish in a clean environment

☐ All required files (datasets, helpers, models) included — code runs without errors

☐ Video demo included (.mp4) with narration, live "Run," and all main sections explained

☐ Everything uploaded directly to Canvas (no external links)

☐ Folder labeled exactly right (PA#-DTSC4525-Firstname Lastname or M#-DTSC4525-Firstname Lastname)

☐ [FOR GRADING] or [FOR PRACTICE ONLY] tag in the submission comments (programming assignments)

Most Frequent Issues That Prevent Review

1. **Silent videos** — narration is required.
2. **Skipping "Run"** — always show the program running live.
3. **Missing datasets/helper files** — if it doesn't run, it can't be reviewed.
4. **Broken environments** — test in a clean environment first.
5. **Wrong formats** — no .rar, .7z, or external links; always a .zip uploaded to Canvas.
6. **Incorrect folder naming** — follow the exact convention.

⇨ Final Project Submission Guidelines

Your final project is worth **50% of your course grade**, divided into milestones to keep you on track (see the milestone table for all deadlines — each with 3 free late days).

The **Final Submission (M4)** is a single zipped folder containing **all** components below.

M4 deadline: All components — written report, code, datasets, slides, and code-demo video — are due together: Mon, Nov 30, 11:59 PM CT (late window ends Thu, Dec 3). You will present LIVE in class during our final meeting: Mon, Nov 30, 3:30–6:20 PM — come ready to present, then submit the full M4 package that same night by 11:59 PM. The live showcase is a scheduled class event and cannot be taken late — if you have a documented conflict, contact me well in advance.

Cover-page requirement: Every milestone report must include a cover page listing the milestone number & title, course number, and **all team members' full names**.

⚠**Important Rule:** If any required component is missing or incomplete, the entire project receives a **zero**. Check carefully before submitting.

Final Written Report

- Written in **Word (.docx) or LaTeX** (template provided) and **exported to PDF** for submission.
- Must be complete, properly formatted, and demonstrate **depth of analysis, critical thinking, and academic writing structure** — and meet the $\geq$ **9,000-word minimum** (excluding references, tables, figures, and supplementary materials).

Think of this as your "mini research paper." Clear structure and polished writing matter as much as technical results — and if you're on a publication track, this document literally becomes your submission draft.

Presentation

- Submit your **PowerPoint (.pptx)** slides **with your M4 folder**, then deliver your presentation and demo LIVE during our final class meeting (Mon, Nov 30) — have your slides presentation-ready at class time; submit the final copies with your M4 folder that night. Have a recorded backup of your demo ready — live AI demos have a sense of humor.
- Your presentation should summarize: **problem & motivation** $\rightarrow$ **methods & datasets** $\rightarrow$ **key results (figures/tables)** $\rightarrow$ **lessons learned & future work**.

Slides should complement your narration — keep text light, focus on visuals.

Python Source Code

- Submit .py or .ipynb files containing the **entire project code** — fully functional and **well-commented**.
- Include **all datasets and any files your project depends on**.
- **Your code must run successfully on the instructor's machine.**
- ⚠**Non-running code = zero for the entire project**, even if every other component is present.

Test by re-running start-to-finish in a fresh environment before submitting — same discipline as your programming assignments.

Video Demonstration (Code Walkthrough)

A narrated walkthrough of your project running — your recorded code demo (the live talk happens in class):

- Show yourself clicking **"Run"** and executing the code.
- Walk through each major part of the code; explain results, outputs, and plots.
- **Narration required** — no silent videos.
- Same policies as programming assignments: missing narration, skipping "Run," or incomplete walkthrough = **zero**.

Imagine you're presenting at a conference: clear, confident, complete.

No External Links

- No Google Drive, Dropbox, GitHub, or other external links — **everything uploads directly to Canvas**.

Labeling Your Final Submission

Label: M4-DTSC4525-Firstname Lastname *(copy-paste and replace the name — space between first and last, no hyphens or merged names; mislabeling = 25% deduction)*

Example — M4-DTSC4525-Mary Jones.zip unzips to:

M4-DTSC4525-Mary Jones/

|— M4-Report-DTSC4525-Mary Jones.pdf (final written report)

|— M4-Code-DTSC4525-Mary Jones.ipynb (or .py — full project code)

|— dataset.csv (+ any other required files)

|— M4-Slides-DTSC4525-Mary Jones.pptx (presentation slides)

'— M4-Demo-DTSC4525-Mary Jones.mp4 (narrated code walkthrough/demo)

(Teams: as with all milestones, each member submits an identical copy under their own name, with all team names on the cover page and in code headers.)

✓ Submission Tips

1. Read and understand all requirements before starting the packaging.
2. Follow the formatting and required file types exactly.
3. **Open and check your files/folders before uploading** — verify everything runs and plays.
4. **Don't wait until the last hour** — technical and internet issues at 11:55 PM are real, and the late window exists for planning, not for living dangerously.

Milestone Progress Reports (M1–M3)

For each non-final milestone, also include a **short Word document** in your zipped folder that:

- **Lists** what the milestone contains,
- **Explains** the progress made and current status,
- **Highlights** any challenges faced or changes from the original plan.

Example *(inside M3-DTSC4525-John Smith)*:

M3 — Implementation & Complete Writeup — DTSC 4525 — John Smith (team: John Smith, Jane Doe)

- Included modules: preprocessing, model training, evaluation
- Status: Initial training complete; hyperparameter tuning in progress
- Issues: Dataset imbalance → will apply SMOTE

This keeps progress visible to both you and me over time — and honestly, these little status notes are excellent practice for industry stand-ups and research lab meetings.

Reminder — milestone deadlines (each +3 free late days): M1: Tue, Sep 8 · M2: Mon, Oct 5 · M3: Wed, Oct 28 · **M4 (all components): Mon, Nov 30**.

☞ Optional Bonus Verbal Exams (Up to +10%)

The course offers up to **two optional in-person verbal (oral) exams, each worth +5%**. These aren't tests to catch you out — they're structured practice for the technical interviews, research meetings, and design reviews waiting for you after graduation, with extra credit as the reward.

Participation is completely optional. You can earn the maximum course grade without taking any verbal exam.

Why Take One?

- Practice interview-style technical explanations in a safe setting
- Build confidence talking through complex ideas under light pressure
- Reinforce deep conceptual understanding — not memorization
- Experience a low-stress, supportive alternative to traditional exams

The focus is on **communication and reasoning**, not on catching mistakes.

Format & Environment

- **Format:** Live, one-on-one conversation with Dr. Aledhari
- **Delivery:** In person — in the classroom (before/after class) or in my office (Discovery Park, Room E292G), based on scheduling
- **Duration:** Typically 10–20 minutes
- **Tone:** Supportive, professional, conversational — modeled on technical interviews, research discussions, and design reviews, *not* high-pressure timed exams

What You Might Be Asked

- Explain a concept or model in plain language
- Describe *why* a technique works and *when* to use it
- Discuss tradeoffs, limitations, or ethical considerations
- Connect lecture concepts to your assignments or project
- Walk through a system design or workflow at a high level

No trick questions. No formula memorization. If you can explain an idea clearly to another person, you're prepared.

Coverage

- **Exam 1:** Lectures 1–6
- **Exam 2:** Lectures 7–12

Specific topics and example questions will be posted on Canvas ahead of time.

How to Sign Up (Required Notice)

- Notify Dr. Aledhari **at least two (2) weeks in advance**.
- Sign-up instructions and time slots will be posted on Canvas.
- Late sign-ups will not be accommodated, and there are **no make-up bonus exams** — the two-week notice is what makes fair scheduling possible for everyone.

Credit Rules

- Take one exam (+5%) or both (+10%).
- Bonus credit is added on top of required components, subject to the Grade Cap & Extra Credit Policy below.
- Bonus exams **do not** replace required work, waive milestones, or excuse missing components.
- Professional communication and academic integrity standards apply.

How to Prepare

Review lecture materials · practice explaining concepts **aloud** (seriously — it's different from thinking silently) · focus on understanding over memorization · connect ideas to real-world and project contexts.

♠ All Bonus Opportunities — One-Page Summary

Everything below is **completely optional** — these exist for flexibility, skill-building, and résumé-strengthening, never as hidden requirements.

#	Opportunity	Bonus	Snapshot
1	**Bonus Verbal Exams**	up to +10%	Two conversational online exams (+5% each); interview practice with a grade boost. Full details above.
2	**Conference Publication Track**	+25%	Submit your class project to a conference and participate. Real conference experience and a strong résumé line.
3	**Journal Publication Track**	up to +50%	Publication-oriented project under Dr. Aledhari's direct supervision. Weekly meetings with demonstrable progress required; inconsistent progress cancels the track. High-impact research experience.
4	**Combined Conference + Journal Track**	up to +60%	One well-executed project powering both submission efforts — maximum professional and academic impact. Requires declaration and strong time commitment.

Tracks 2–4 are alternatives (pursue at most one); verbal exams may be added on top of any path. All publication tracks require project completion and Dr. Aledhari's approval, and projects must still meet every course milestone.

Important Reminders

- Bonus opportunities **do not** replace required coursework, waive milestones or deadlines, or excuse missing components.
- **Grade Cap & Extra Credit Policy:** Extra credit may raise your numerical grade above 100%, but letter grades are capped per university policy.
- You **must declare bonus-track intent** via the Canvas project questionnaire so proper guidance can be provided.

Bottom Line

Bonus options give you choices — interview practice, conference exposure, research experience, or simply grade cushion. Pick what serves your goals, or none at all; the required work done well is a complete path to the top grade.

▲ Grading Scale

Percentage	Letter Grade	What It Means	What It Looks Like
90 – 100%	A (Excellent)	Exceeds expectations — deep understanding, creativity, and strong application of concepts. Goes well beyond the basics.	Consistently submits polished programming assignments with features beyond requirements, implements a novel twist in the final project, and provides insightful analysis in the optional verbal exams.
80 – 89%	B (Good / Above Average)	Solid work that fully meets course expectations — clear grasp of fundamentals, applied to moderately advanced problems.	Submits all assignments correctly, though solutions are mostly straightforward. Project is functional and well-structured but lacks originality or deeper analysis.
70 – 79%	C (Satisfactory / Average)	Meets minimum course requirements — basic understanding, but struggles with advanced applications or integration. **This is the grade most degree plans require in major courses (see the Undergraduate Standing Note below).**	Completes assignments with correct solutions but limited explanation. Project works but has missing components or weak analysis.
60 – 69%	D (Below Average)	Meets some but not all requirements — partial understanding with major gaps in concepts or execution. **At UNT a D earns credit hours, but many degree plans require a C or better in major and prerequisite courses — so a D here may not count for your program. Check your degree audit.**	Misses an assignment, struggles with project implementation, or misapplies models. Shows effort but lacks full comprehension.
0 – 59%	F (Failing)	Does not meet minimum expectations — insufficient understanding or incomplete submission of required components.	Multiple missed assignments, incomplete project, or lack of engagement with course material.

Undergraduate Standing Note: Per UNT academic standards, undergraduate students must maintain a minimum cumulative GPA of **2.0** to remain in good academic standing; falling below 2.0 places you on academic probation, and continued deficiency can lead to academic suspension. A grade of **D** earns credit hours at UNT, but **many degree plans require a C or better in major and prerequisite courses** — confirm with your degree audit or academic advisor. In practical terms: aim for C or above to protect your degree requirements, and B or above to keep your GPA comfortably safe — and if you find yourself heading toward D territory, come talk to me *early* in the semester, not in December. There is almost always a recoverable path if we catch it in time.

Rounding Rule (in your favor)

Final grades are **rounded up when the decimal is 0.5 or higher**:

Final Score	Letter
89.5	A ✓
89.4	B
79.5	B ✓
79.4	C

This keeps grading consistent and fair for everyone — and please don't email in December asking for the 89.4 → A bump; the rounding rule *is* the flexibility, applied identically to all.

Materials & Technology

The Short Version: This Course Costs You (Almost) Nothing

- **Required textbook: None.** Zero purchase required.
- **All software: Free.** Every tool we use is open source or has a free tier sufficient for coursework.
- **GPU compute: Free.** No GPU purchase or cloud bill needed — free platforms cover everything (details below).
- **APIs:** Some assignments use LLM APIs. Free-tier options exist for nearly everything; if you choose paid APIs, expect roughly **$0–$20 total** for the semester depending on your project.

Recommended (Optional) Readings — For Going Deeper

This is a hands-on, project-focused course preparing you for both industry and research. Generative AI evolves so fast that no single textbook covers our material — so we'll work from published research papers, tutorials, and technical resources posted on Canvas each week.

That said, if you want a reference library, these are excellent. **None are required**, and most are free to access:

Free to read online:

1. **Speech and Language Processing** (3rd ed. draft) — Dan Jurafsky & James H. Martin. *The* classic NLP text — the full draft is freely available on the authors' Stanford webpage.

Commercial titles — Good news: UNT gives you free access to the O'Reilly Learning Platform through the university library. That means every O'Reilly title below (#2, 4, 6, 7, 8, 9, 11) is **free to read online** with your UNT login — no purchase needed. Log in via the UNT Libraries website to get started.

2. **Hands-On Large Language Models** — Jay Alammar & Maarten Grootendorst. O'Reilly, ISBN 9781098150969 (2024)
3. **Build a Large Language Model (From Scratch)** — Sebastian Raschka. Manning, ISBN 9781633437166 (2024) — *pairs beautifully with our micro-series*
4. **Natural Language Processing with Transformers** (Rev. 1st ed.) — Tunstall, von Werra & Wolf. O'Reilly, ISBN 9781098136796 (2022)
5. **Transformers for Natural Language Processing and Computer Vision** (3rd ed.) — Denis Rothman. Packt, ISBN 9781805128724 (2024)
6. **AI Engineering: Building Applications with Foundation Models** — Chip Huyen. O'Reilly, ISBN 9781098166304 (2025)
7. **LLMOps: Managing Large Language Models in Production** — Abi Aryan. O'Reilly, ISBN 9781098154202 (2025)
8. **Generative Deep Learning** (2nd ed.) — David Foster. O'Reilly, ISBN 9781098134181 (2023)
9. **Essential Math for AI** — Hala Nelson. O'Reilly, ISBN 9781098107635 (2023) — *great if you want a math refresher*
10. **Mathematics of Machine Learning** — Tivadar Danka. Packt, ISBN 9781837027873 (2025) — *ditto*
11. **Building Applications with AI Agents** — Michael Albada. O'Reilly, ISBN 9781098176501 (2025)

Not sure where to start? For most students: #2 for the big picture, #3 if you love building from scratch, #6 if you're industry-focused, #9 or #10 if the math prerequisites feel rusty.

Hardware & Internet

	Requirement
Computer	Laptop or desktop that runs Python — minimum **8 GB RAM, 16 GB recommended**. No GPU needed: heavy compute runs on free cloud platforms (see Free GPU Access below).
Internet	Reliable broadband for downloading datasets, calling APIs, and streaming lecture videos.
Headset & webcam	Needed only for **optional** live activities you opt into — bonus verbal exams and office-hours Teams/Zoom calls. Nothing graded requires being on camera.

If your computer can't meet these requirements, contact me in Week 1 — free cloud platforms plus campus lab machines cover everything.

Software & Development Tools — All Free

Category	Tools
Language	Python 3.9+
Environments	Jupyter Notebook, VS Code, or Google Colab (free tier handles most tasks)
Deep learning	PyTorch (preferred) or TensorFlow · Hugging Face Transformers
Vector databases (RAG)	FAISS (open source) · Pinecone (free academic tier) · Weaviate (open source / free cloud tier)
Version control	GitHub — free account recommended for version control and submissions
Documents	Overleaf (LaTeX, free tier) · Microsoft Office — **free for all UNT students via Office 365** (download or use online with your UNT account)

Setup help: A step-by-step environment setup guide (with a verification notebook to confirm everything works) will be posted on Canvas in Week 1. If anything won't install, ask on the discussion board or come to office hours — don't lose a weekend to a setup problem.

Free GPU Access — When Your Laptop Isn't Enough

Some assignments and projects (especially fine-tuning in PA3 or ambitious final projects) benefit from GPU acceleration. **You never need to buy hardware or pay for cloud compute in this course.** Here are your free options, in the order I recommend trying them:

Free Notebook Platforms — Your Practical Default

These give you a full environment with standard ML libraries pre-installed, and **none require a credit card**. Start here:

Platform	Hardware	Free Allowance	Things to Know
Kaggle Notebooks	P100 (16 GB) or dual T4 (32 GB combined); TPU v5e-8	~30 GPU-hrs/week (quota counter resets on a fixed cycle); ~20 TPU-hrs	12-hr session cap · 20 GB persistent storage · phone verification required · enable "Internet" in settings before pip install
Google Colab	T4 (~15–16 GB)	Not officially published; varies with demand (commonly ~15–30 GPU-hrs/week)	12-hr session cap · idle disconnect after ~90 min · no background execution on free tier · disk is ephemeral, so mount Google Drive · may drop to CPU at peak times
Lightning AI	T4-class, single GPU	15 monthly credits (~22 T4 GPU-hrs); credits don't roll over	Studios restart every 4 hrs · no A100/H100-class GPUs on free plan · 100 GB Studio + 10 GB Drive storage
Saturn Cloud · Paperspace Gradient · Intel Tiber AI Cloud	Mid-range GPU (typically 8–16 GB)	Standing free tiers, each with its own quota	Great overflow options when your Kaggle and Colab quotas run out
AWS SageMaker Studio Lab	Free GPU, no card	—	⚠ Closed to new sign-ups as of July 30, 2026; existing accounts still work. Only relevant if you already have one

Skip the GPU Entirely — Hosted Model APIs

For many tasks, you don't need your own GPU at all: services like **NVIDIA NIM, Google AI Studio, Groq Cloud, and Cerebras** offer generous free-tier access to hosted open-weight language models. If your task is *using* a model rather than *training* one, this is often the fastest path.

Practical Tips

- **Plan around quotas:** Free GPU hours are weekly/monthly budgets. Prototype on CPU or a small data sample first, then spend GPU time on full runs.
- **Save your work constantly:** Free sessions can disconnect. Checkpoint models and sync outputs to Drive/GitHub so a disconnect costs you minutes, not days.
- **Stack platforms:** Kaggle + Colab + Lightning together give you 60+ free GPU-hours per week — more than any assignment in this course needs.
- **Stuck without compute near a deadline?** Email me *before* the deadline, not after. We'll figure it out.

A short setup walkthrough video for each platform will be posted on Canvas.

External APIs & Cloud Services

Some assignments (e.g., PA6, PA7, PA8) require LLM API access. You'll create your own accounts and API keys — this is a genuinely valuable professional skill, and Canvas will include a guide for doing it safely.

Keeping this free (or nearly free):

- **Google AI Studio (Gemini)** offers a genuinely free API tier — no credit card required — making it the workhorse option for coursework.
- **GitHub Models** is free with any GitHub account and exposes multiple major model families — a great backup.
- **OpenAI and Anthropic** provide small starter credits rather than ongoing free tiers — fine for trying a specific model, and assignments will always list a free-tier path.

- **Golden rules:** never hardcode API keys in submitted code (use environment variables — we'll show you how), never share keys, and set a spending cap on any account with a payment method attached.

Bottom line: you can complete every assignment in this course for **$0** using free tiers. Paid APIs are optional conveniences, not requirements.

Latest LLM Models & Estimated API Costs (Verified August 2026)

⚠**This field moves fast.** These prices were verified in August 2026, but providers change models and rates frequently — sometimes monthly. **Always check the official pricing links before spending money.** If you spot outdated info here, tell me and earn my gratitude (and help your classmates).

OpenAI — GPT Family

Current flagship lineup: The GPT-5.6 family (Sol, Terra, Luna) went generally available on July 9, 2026, with a ~1M-token context window across all three tiers.

Model	Input (per 1M tokens)	Output (per 1M tokens)	Best For
GPT-5.6 Sol (flagship)	≈ $5.00	≈ $30.00	Hardest reasoning & coding tasks
GPT-5.6 Terra	≈ $2.00	≈ $12.00	Strong all-purpose workhorse
GPT-5.6 Luna	≈ $0.20	≈ $1.20	High-volume, cost-sensitive work — *your default for assignments*

Cost savers: the Batch API is 50% off, and cached input is 90% off on the GPT-5.6/5.5/5.4 families. Note that long prompts cost more: above the short-context threshold, Sol rises to $10/$45 and Terra to $4/$18.

Official pricing: https://openai.com/api/pricing

Anthropic — Claude Family

Current lineup *(refreshed substantially in mid-2026)*:

Model	Input (per 1M tokens)	Output (per 1M tokens)	Best For
Claude Fable 5 (frontier flagship)	≈ $10.00	≈ $50.00	Most demanding reasoning & agentic work
Claude Opus 5	≈ $5.00	≈ $25.00	Premium reasoning & complex coding
Claude Sonnet 5	≈ $3.00	≈ $15.00	Best price/performance — *your default for assignments*
Claude Haiku 4.5	≈ $1.00	≈ $5.00	High-volume, low-cost production work

Sonnet 5 has introductory pricing of $2/$10 through August 31, 2026, with standard $3/$15 pricing starting September 1 — I've listed the standard rate since our semester runs past that date. The Batch API gives a 50% discount, and prompt caching saves up to 90% on cached input.

Official pricing: https://platform.claude.com/docs/en/about-claude/pricing

Google — Gemini Family

Current lineup:

Model	Input (per 1M tokens)	Output (per 1M tokens)	Best For
Gemini 3.1 Pro (≤200K context)	≈ $2.00	≈ $12.00	Frontier reasoning, long documents

Model	Input (per 1M tokens)	Output (per 1M tokens)	Best For
Gemini 3.1 Pro (>200K context)	≈ $4.00	≈ $18.00	Very long-context workloads
Gemini 3.6 Flash	≈ $1.50	≈ $7.50	Fast workhorse — launched July 21, 2026 with a 17% output price cut vs. 3.5 Flash
Gemini 3.1 Flash-Lite	≈ $0.25	≈ $1.50	High-volume budget work

Most important for you: Google AI Studio's free tier remains the best free option for coursework — Flash models are free at 5–15 requests/minute with up to 1,000 daily requests. Note that Pro models were removed from the free tier on April 1, 2026, so plan Pro experiments on the paid tier or use Flash. A unique Gemini advantage: every model, down to the cheapest Flash-Lite tier, supports a 1M-token context window.

Official pricing: https://ai.google.dev/gemini-api/docs/pricing

Other APIs & Tools (Optional / Assignment-Dependent)

These are complementary — not required — and useful for building apps, analyzing text, or prototyping. Many have free tiers.

Hosted / API tools:

- **Hugging Face Inference Providers** — a unified, OpenAI-compatible gateway routing to 15+ providers (Groq, Together AI, Fireworks, Replicate, Cerebras, Cohere, and more), with pass-through pricing at the underlying provider's rate. Free accounts get small monthly experiment credits; PRO ($9/mo) includes $2.00 in monthly credits plus a bigger ZeroGPU quota. Great for open-model experimentation.
- **Groq Cloud / Cerebras / NVIDIA NIM** — very generous free tiers for blazing-fast hosted open-weight models (see our Free GPU Access section).
- **Cohere API** — Command models with strong RAG and tool use; trial credits, then billed usage.
- **Together AI / Fireworks / Replicate** — open-model hosting with pay-per-use pricing; Replicate is handy for one-off runs of image/audio models. *(Note: Banana.dev from older guides no longer operates — use these instead.)*
- **Stability AI** — image and multimodal models with demo tiers and paid APIs.
- **ElevenLabs** — text-to-speech / voice API; free tier (10K characters/month) with paid plans from $5/month. Useful for PA7 (multimodal).

Key libraries & local-first tools (no API billing) — run locally or in your free cloud environment:

Category	Tools
LLM orchestration & tooling	LangChain, LlamaIndex, Haystack
Embeddings & vector search	SentenceTransformers, FAISS, Annoy
ML foundations	Pandas, NumPy, scikit-learn
Visualization / analysis	Matplotlib, Seaborn, Plotly
Local LLM runners	**Ollama, LM Studio** — run open-weight models entirely on your own machine, $0 forever *(new addition — these have become standard student tools)*

Tip: Combine local libraries with cloud APIs to optimize costs — do heavy preprocessing locally, and only the final API calls incur token charges.

Practical Student Notes

Billing basics

- API usage is billed separately from course materials; you're responsible for your own API costs.
- Free tiers cover most coursework (Google AI Studio, GitHub Models, Groq, Hugging Face) — heavy usage is what triggers billing.
- Token cost = input + output. Always count both when estimating. **Watch out for reasoning models:** "thinking" tokens bill as output even though you never see them — a single hard question can cost far more than its visible answer suggests.

Cost-optimization tips

- **Cache results** to avoid redundant API calls.
- **Default to the cheap tier:** GPT-5.6 Luna, Claude Haiku 4.5, or Gemini Flash handle most assignment tasks; save flagships for when quality measurably matters — comparing them is itself a great experiment for PA4.
- **Use Batch APIs** — all three major providers offer ~50% off for non-urgent jobs.
- **Monitor your usage dashboard** with each provider, and set spending caps/alerts on day one.

Educational credits — Some providers offer academic credits via individual application (e.g., GitHub Student Developer Pack unlocks several AI perks). Worth 15 minutes of sign-ups at semester start.

Choosing the Right Model (Practical Cheat Sheet)

Task Type	Recommended Model Class
Coding & logic-heavy tasks	GPT-5.6 Sol/Terra · Claude Opus 5
Complex reasoning & agentic workflows	Claude Fable 5 / Opus 5 · GPT-5.6 Sol
Everyday assignment work (best value)	GPT-5.6 Luna · Claude Sonnet 5 · Gemini 3.6 Flash
Very long context (huge docs, codebases)	Gemini 3.1 Pro (1M ctx on all tiers) · Claude / GPT-5.6 1M-ctx models
Cost-sensitive & experimentation	Gemini free tier · GitHub Models · Groq · open models via Hugging Face · Ollama (local, $0)
Speech / voice integration	ElevenLabs · OpenAI audio APIs
Image generation	Stability AI · Gemini image models · open models via Replicate

Official Pricing & Docs (Live Links — always check these first)

Provider	Pricing URL
OpenAI	https://openai.com/api/pricing
Anthropic Claude	https://platform.claude.com/docs/en/about-claude/pricing
Google Gemini API	https://ai.google.dev/gemini-api/docs/pricing
Hugging Face	https://huggingface.co/pricing
Cohere	https://cohere.com/pricing
Groq	https://groq.com/pricing
ElevenLabs	https://elevenlabs.io/pricing

☞ Policy on Generative AI & LLMs

You are **encouraged** to use AI tools in this course — how could we teach generative AI and ban it? But use them as **assistants, not replacements** for your own thinking and work.

✓ **Encouraged uses:**

- Brainstorming project ideas
- Clarifying concepts you're stuck on
- Debugging code with AI suggestions
- Polishing grammar or wording

Not allowed:

- Submitting AI-generated work without your own effort and understanding
- Copying code or essays directly from AI
- Relying on AI output without verifying its accuracy

Disclosure rule: If you use AI to help with your work, include a short note in your submission (e.g., *"Used Claude to brainstorm experiment design"* or *"Used ChatGPT to debug the attention mask function"*). Honest disclosure is never penalized — it's professional practice.

The understanding test: Whatever you submit, you should be able to explain — line by line, decision by decision. *(The optional verbal exams are a great way to prove this to yourself.)* I may also ask you to walk me through your code in class or office hours — this is normal, not an accusation.

▲ Cost & Licensing Disclaimer

This course involves third-party APIs (OpenAI, Anthropic, Google, Hugging Face, Cohere, Stability AI, ElevenLabs, and others). While many offer free tiers or trial credits, **you are responsible for:**

- Creating and managing your own accounts & API keys
- Any costs beyond free quotas
- Monitoring your usage to avoid surprise charges

Realistic cost estimate: $0–$20 for the semester. Every required assignment has a free-tier path (typically Google AI Studio, GitHub Models, or Groq); spending money is a convenience choice, not a requirement. Neither the instructor nor the university can provide credits, API keys, or reimbursements — so set spending caps on day one.

✉ Course Policies & Communication

Attendance & Participation

Attendance is expected at every class meeting. Our sessions are hands-on — live coding, discussion, immediate help — and what happens in the room is genuinely hard to replicate alone, so I take attendance each week. Students who attend consistently earn higher grades with less stress; that pattern holds every semester.

Your absence allowance: you may miss up to **two (2) classes** with no penalty and no note needed — but treat these as your **emergency reserve** (illness, family matters, the unexpected), not casual skips. **Each absence beyond the second incurs a 5% deduction from your final course grade** (3rd absence = −5%, 4th = −10%; your 5th absence triggers the rule below). If you do miss, check Canvas for the slides and materials, and ask a classmate for notes — and for extended situations, email me early so we can make a plan together.

▲**Five absences = F for the course:** if your total absences reach **five (5) class meetings** — counting **both** your two free absences **and** any penalized ones — you receive an **F** for the **entire course**, regardless of your grades on all other requirements and bonuses. **If a job, other commitments, course conflicts, or any other circumstance means you cannot reliably attend in person by or before class time, you are strongly encouraged to move to the online section (DTSC 4525-401), which has no attendance requirement.**

University-excused absences (religious holy days, military service, and other absences covered by UNT Policy 06.039) do not count toward your two-absence allowance or the five-absence limit — notify me in advance where the policy requires it, and affected work can be completed without penalty.

The one attendance-critical date: the LIVE showcase during our final class meeting (Mon, Nov 30) cannot be made up — if you have a documented conflict, contact me well in advance.

Professional Communication Guidelines

Effective, respectful communication creates a positive learning environment — and between class meetings, email *is* our hallway. I commit to communicating with you professionally, supportively, and respectfully, and I ask the same of you.

Think of every email you send as professional practice: the habits you build here are the same ones you'll use with managers, collaborators, and clients.

Respectful Communication

- Be courteous and respectful in all emails and messages.
- Disrespectful or inappropriate communication will not receive a response.

Email Etiquette

Start with a proper greeting, addressing me formally:

- "Dear Dr. Aledhari"
- "Hello Professor"
- "Hi Dr. Aledhari"
- "Good Morning Dr. Aledhari"

Emails without a proper greeting may not receive a reply.

Structure every email in four parts:

1. **Subject line** — clear and specific, and include the course number (e.g., "DTSC 4525 — Question about Project Milestone 2").
2. **Greeting** — respectful (examples above).
3. **Message** — state your question or concern clearly and concisely.
4. **Closing** — "Best regards" or "Sincerely," followed by your **full name** (this matters — I can't look up "Mike" in a gradebook).

General guidelines:

- **Clear & concise** — short and to the point beats long and winding.
- **Grammar & spelling** — proofread before sending.
- **No slang or texting abbreviations** — "you," not "u"; skip "wassup."
- **Formatting** — readable fonts (12–14 pt); no ALL CAPS (it reads as shouting).
- **Emoticons & humor** — sparingly; tone is easily misread in text.
- **Privacy** — never include sensitive personal information in email.

Examples of Well-Written Emails

Example 1 — Asking about an assignment:

Subject: DTSC 4525 — Clarification on Programming Assignment 2

Dear Dr. Aledhari,

I am working on Programming Assignment 2 and am unsure about the evaluation criteria for model performance. Could you please clarify whether we should report accuracy only, or include precision and recall as well?

Thank you for your time and guidance.

Best regards, Jane Smith

Example 2 — Signing up for a bonus verbal exam:

Subject: DTSC 4525 — Sign-Up for Bonus Verbal Exam 1

Hello Professor,

I would like to sign up for Bonus Verbal Exam 1 covering Lectures 1–6. I understand the two-week notice requirement and have reviewed the available time slots on Canvas. My top three preferences are [slot 1], [slot 2], and [slot 3].

I appreciate your time.

Sincerely, John Doe

❤ Professionalism, Collaboration & Group Work

Working together is a big part of learning in this course. To keep group projects fair, respectful, and productive:

Expectations

- **Respect:** Treat all team members with professionalism — listen actively, value others' contributions, avoid dismissive behavior.
- **Deadlines:** Agree on **internal team deadlines ahead of the official ones** so last-minute rushes don't create tension (your 3 late days are a shared safety net, not a plan).
- **Roles:** Assign roles (coordinator, coder, tester, writer) to balance workload — and **rotate them across milestones** so everyone grows.
- *Example: For M2, John leads coding, Maria writes, Aisha coordinates. For M3, roles shift.*

Conflict Resolution — the Step-by-Step Process

Teams hit bumps — that's normal. Here's the path:

1. **Talk first:** Raise concerns respectfully within the team.
2. **Document:** Keep notes of what was agreed (tasks, deadlines, issues).
3. **Ask for mediation:** If problems continue, notify me **early** — I can meet with the group (at office hours or by appointment) to help resolve things.
4. **Final step:** If a member consistently doesn't contribute despite these efforts, the group may recommend grade adjustments, and together we decide what's fair.

Example: A member repeatedly misses team meetings → the team documents it, discusses it directly, then informs me. Early beats December, every time.

Why This Matters

Collaboration is a skill employers weigh as heavily as technical ability. These habits — clear roles, documented agreements, early escalation — are exactly how strong engineering teams operate.

❄ Project Teamwork & Submission Policies

Team Size

- Work **individually or in a team of up to 3 students**.
- All members must be officially enrolled in DTSC 4525.
- Collaboration with students from other courses or outside groups is **not allowed**.

Individual Submission with Team Credit

- As established in the submission rules: **each member submits an identical copy** of every milestone under their own name.
- Every milestone report must include a **"Team Contributions" section** — identical in all copies — listing each member's specific contributions (e.g., "John: data pipeline & training loop; Maria: related work & results analysis; Aisha: RAG integration & experiments").
- This ensures everyone's efforts are visible and credited while keeping submissions consistent.

Group Composition

- Every report lists **all team members' names** (cover page + code headers).
- Each student submits under their own name in Canvas.

Exclusivity

- Your project must be **original work created for this course**. Reusing projects from, or sharing across, other classes results in a **zero** for the project.

Team Changes

- If teammates drop the course, you may need to continue solo — the milestones and requirements stay the same, so **plan ahead and keep backups** of all shared work (a shared repo or drive folder from Day 1 is cheap insurance).

Non-Participation Reporting

- If a teammate isn't contributing, **notify me as soon as possible** — fairness is taken seriously, and individual grade adjustments may be made.
- *Good communication usually prevents problems before they grow. Teams that set roles early, divide tasks, and check in weekly almost always outperform.*

■ Digital Privacy & Third-Party Tools

Your privacy and security matter. Here's what we use and what you should know:

Tools We Use

- **Canvas (LMS)** — home for all grades, assignments, announcements, and feedback.
- **Microsoft Teams/Zoom** — virtual office hours by arrangement, and remote backup for class meetings when needed.
- **Turnitin** — integrated into Canvas to check originality of written work, protecting academic integrity.
- **PollEverywhere (or similar)** — used occasionally for live polls, quick comprehension checks, and feedback during class.

Data & Privacy Notice

- These tools may collect basic personal information (name, email, participation records).
- **Grades live only in Canvas** — never in third-party tools.
- Teams/Zoom sessions may include chat logs; **recordings are only made with advance notice**.
- Turnitin stores submissions securely, solely for cross-institution originality comparison.

Your privacy matters. If you have concerns about any tool, talk to me — solutions are always available.

▼ University Policy Statements & Student Resources

Academic Integrity

Academic honesty is essential to your success and to the integrity of UNT. All work you submit must be your own unless collaboration is specifically allowed. Cheating, plagiarism, or any form of dishonesty carries serious consequences — including a possible zero — as outlined in the **UNT Academic Integrity Policy** (see the policy at UNT's Academic Integrity page).

In this course specifically:

- **Programming assignments are individual.** Discussing approaches is encouraged; submitting someone else's code (from a classmate, a website, or an AI tool without disclosure) is plagiarism.
- **Projects may use outside materials** (images, code snippets, papers, slides) — **with credit to the original source, always**.
- **AI tools:** allowed as assistants with disclosure — see the *Policy on Generative AI & LLMs* section. Undisclosed or wholesale AI-generated work is an integrity violation.
- **Team submissions must be truly identical** across members (see submission rules).

Citing sources isn't just about avoiding penalties — it shows professionalism, respect, and academic maturity. Unsure whether something counts as acceptable collaboration? Ask me first — that question is always welcome and never penalized.

Accessibility & Accommodations (ODA)

UNT makes reasonable accommodations for students with disabilities through the **Office of Disability Access (ODA)**:

Sage Hall, Room 167 · (940) 565-4323 · studentaffairs.unt.edu/office-disability-access

How it works: Register with the ODA (application + documentation, each semester) → once eligibility is confirmed, request your **Letter of Accommodation** → ODA emails it to me → **then reach out so we discuss your needs privately** (a Teams/Zoom office-hours call or email works great). You can register anytime, but **earlier is better** — accommodations aren't retroactive, so contacting ODA in Week 1 protects your whole semester.

This course was designed with accessibility in mind — recorded lectures you can rewatch at any speed, flexible deadlines with built-in late days, and screen-reader-conscious materials. If anything is still a barrier, tell me and we'll fix it.

Title IX & Sexual Misconduct

UNT is committed to a safe learning environment; sexual harassment, discrimination, and misconduct are prohibited. **I am a mandatory reporter** — I must report incidents of sexual misconduct shared with me to the Title IX office.

TitleIX@unt.edu · (940) 565-2759 · UNT Title IX Resources

Prefer confidentiality? Counseling & Testing Services (below) is a confidential resource that does *not* trigger a report.

Privacy & Recording (FERPA)

- Your grades and personal information are protected under **FERPA** and will never be shared without your consent.
- Course materials, posted recordings, and other students' contributions **may not be shared outside this class** — you may rewatch a posted lecture on Canvas, but you may not post it to social media or share it externally.
- **Students may not record one-on-one meetings (Teams/Zoom office hours, verbal exams) without my prior approval.** Unauthorized recording violates privacy expectations and university policy.
- If I record any live interaction, you'll be informed in advance, and recordings are used only for educational purposes.

Communication & Feedback

Contacting me: Email is the preferred method (clear subject line with "DTSC 4525" — see the Email Etiquette section). I typically reply **within 24–48 hours on workdays**; weekend/holiday emails get answered the next business day. During busy periods my inbox fills up — **if you haven't heard back in two business days, please send a follow-up. A gentle nudge is always appreciated, never rude.**

Announcements: All updates, deadline reminders, and notices post to **Canvas Announcements**. Check Canvas **at least twice per week** — and turn on notifications so you never have to remember to.

Feedback timelines:

- Assignments & milestones: grades and feedback **within 7–14 days** of submission — milestone feedback is timed to guide your next step before the following deadline.
- Drafts & questions: I'll give brief guidance on drafts anytime; detailed grading feedback comes after submission.

Safety & Emergencies

- **Eagle Alert** is UNT's emergency notification system — keep your contact info current in myUNT so you receive campus alerts.
- **In crisis?** Call or text **988** (Suicide & Crisis Lifeline, 24/7), or reach UNT Counseling & Testing Services below. If you're ever in immediate danger, call 911.

Student Support & Well-Being

Your success matters — at UNT, help is everywhere and using it is a sign of strength:

Need	Resource
Tutoring & study skills	**UNT Learning Center** — learningcenter.unt.edu
Writing help (any stage)	**UNT Writing Center** — writingcenter.unt.edu (online appointments available)
Research & databases	**UNT Libraries** — library.unt.edu
Tech problems (Canvas, email)	**UNT IT Help Desk** — (940) 565-2324 · helpdesk@unt.edu
Confidential counseling	**Counseling & Testing Services** — (940) 565-2741 · Chestnut Hall 313
Health care	**Student Health & Wellness Center** — (940) 565-2333 · Chestnut Hall, 2nd floor
Food insecurity	**UNT Food Pantry** / Student Success Office — discreet, immediate help
Financial guidance & emergency funds	**Financial Aid** — financialaid.unt.edu
Careers, internships, résumés	**UNT Career Center** — careercenter.unt.edu

Example: struggling to afford meals? Contact the Food Pantry discreetly. Feeling overwhelmed? Counseling Services offers consultations — often same-day.

Talk to Me

My top priority is that you learn a lot and enjoy the journey. If something is helping — or getting in the way — **tell me early**: by email or at office hours, positive or constructive. Don't wait until it's "too late"; if something feels unclear, stressful, or unhelpful, speak up sooner so we can fix it together. You'll also receive the official **SPOT course evaluation** near semester's end — please complete it; I read every response and this course improves because students speak up.

Student Wellness

Taking care of yourself is part of doing well academically:

- Eat balanced meals, sleep well, move your body.
- Take real breaks — rest is when learning sticks.
- Avoid unhealthy coping (skipped sleep, over-caffeinating, substance use).

If you face challenges — academic stress, anxiety, family struggles, financial difficulty — **you are not alone**. Reach out to a friend, to me, or to any campus resource above. Asking for help shows strength, not weakness.

Classroom Behavior

Our discussion boards, emails, and virtual meetings are shared learning spaces governed by the **UNT Student Code of Conduct**. Treat classmates, the instructor, and yourself with courtesy; healthy debate is welcome, personal attacks never are. Violations may lead to disciplinary action — but more importantly, when everyone contributes positively, this course becomes supportive, engaging, and welcoming for all.

Accessible Course Materials

This syllabus and all course materials are prepared with accessibility in mind: clear structure, accessible headings/lists/tables, screen-reader-compatible files wherever possible, and readable fonts and colors in slides. If any material presents a barrier for you, tell me and I'll provide an accessible alternative promptly.

Syllabus Change Policy

This syllabus is your roadmap. If adjustments become necessary (university calendar changes, a topic needing more time), I'll communicate them **in advance via Canvas announcements and email** — deadlines never move earlier, and changes always aim to support your learning, never to surprise you.

❀ Flexibility & Contingencies

Life is unpredictable — this course is built so your learning doesn't break when plans do.

If Campus Closes

- If the university closes due to weather, public health concerns, or other emergencies, **our class continues online**: materials post to Canvas, and any affected class meeting moves to Teams/Zoom or an asynchronous activity — watch Canvas for instructions. Assignments and due dates stay the same unless I announce otherwise.
- If a **widespread emergency** (e.g., extended power/internet outages across the region) affects many students' ability to work, I'll post adjusted instructions and, if needed, adjusted deadlines on Canvas.

If the Instructor Is Unavailable

- If I'm temporarily out due to illness or emergency, you'll get clear instructions via Canvas: a guest instructor, pre-recorded materials, or an adjusted timeline.
- For anything longer-term, I'll provide an alternate point of contact (such as a department coordinator).

If Technology Fails

- **Canvas down at a deadline?** Email your submission directly to me as an attachment **by the deadline** — timestamped email counts.
- **Teams/Zoom down for a meeting?** Check Canvas announcements or email for the backup plan (rescheduled time or alternative platform).

Example scenarios:

- *Ice storm closes campus on a class day* → we meet via Teams/Zoom at the usual class time, or the session posts asynchronously; watch Canvas.
- *I'm out sick* → a recorded lecture or activity posts, and any affected deadline extends.
- *Canvas is down the night PA5 is due* → email me the zip by 11:59 PM; no penalty.

You are never expected to figure out emergencies alone. Communicate first — if something goes wrong, tell me right away and we'll find a fair solution.

☆ How to Succeed in This Course

Time expectations: Plan on roughly **9 hours per week total** (the standard for a 3-credit course — this assumes the full prerequisite background; your actual hours may be less or more, and both are normal). Spend it on lecture videos and notes, assignments and project work, and supplemental reading. Students who invest consistently, week after week, earn higher grades with less stress — the math never works for cramming a build-heavy course.

Study strategies that work:

- **Review regularly:** 20–30 minutes daily beats a pre-deadline cram.
- **Code often:** programming is learned by doing — small daily exercises sharpen skills fast (the optional micro-series is perfect for this).
- **Use study groups:** explaining a concept to a peer is one of the best comprehension checks there is.
- **Ask early, ask often:** don't marinate in stuck-ness — post on the discussion board, email me, or drop into office hours at the first roadblock.
- **Connect concepts to what you care about:** tying LLMs, RAG, or agents to problems in your own field makes learning stick — and often becomes your project topic.

What success looks like (real patterns from past students):

- **Student A** blocked two evenings weekly for assignments and one weekend morning for the project — finished early, low-stress December.
- **Student B** formed a 3-person virtual study group on Sundays — reviewed notes, swapped debugging tips, practiced explaining concepts aloud (great verbal-exam prep, too).
- **Student C** used short 10–15-minute office-hours check-ins whenever stuck — small questions, fast unblocks, on topics like attention mechanisms.

- **Student D** used the Learning Center to shore up linear algebra and probability early — and cruised through the advanced ML weeks because of it.

Remember: success here isn't about being perfect — it's consistency, questions, and steady progress. Small efforts each week add up to big achievements by December.

❑ Suggested Time Management Plans for the Four Student Scenarios

Four realistic weekly rhythms, matched to the Student A–D patterns from How to Succeed in This Course. Borrow one as-is or mix your own — each is built around our Monday-deadline rhythm and fits within the ~9-hour weekly expectation.

Plan A — The Steady Scheduler (working professional): After Monday's class: review slides & notes (1 h). Wed & Thu evenings: current programming assignment (2 h each). Sat morning: project milestone work (3 h). Sun: 30 min to review the week, plan the next, and check Canvas. Steady, predictable, and finished before every deadline — the late days stay untouched as true insurance.

Plan B — The Study-Group Builder: Mon–Tue: class + notes review. Thu evening: 1-hour group call — debug together and divide milestone tasks. Sun: 2–3 h group session to review the week and practice explaining one concept aloud each (excellent verbal-exam and presentation prep). Individual assignment work fills the gaps. Post in the "Meet Your Classmates" forum in Week 1 to find your group.

Plan C — The Front-Loader: Use the head start: everything is released Aug 17, and the first PA isn't due until Sep 14 — draft two or three assignments during those lighter early weeks. Mid-semester: polish one PA per week alongside milestones. November: full focus on M3–M4, finishing the entire project by Nov 30. December becomes celebration, not crunch.

Plan D — The Foundation-First: Weeks 1–3: add ~2 h/week on math refreshers (the Learning Center, the optional micro-series, or the recommended math texts) before the deep-learning weeks arrive. Choose the more application-focused assignments (PA4–PA8) as your graded five. Keep milestones on schedule and visit office hours every other week — short check-ins compound.

➥ Getting Started Checklist (First 48 Hours)

Welcome! Set yourself up for success by completing these steps within your first 48 hours:

1. ✓ **Log into Canvas**
   - Confirm you can access the course shell, modules, and announcements; bookmark the homepage and **turn on notifications**.
   - *If anything looks missing or broken, contact me or the IT Help Desk (940-565-2324) immediately — don't wait.*
2. ✓ **Download the syllabus**
   - Read the grading policy, submission rules, and course schedule; save a copy locally.
   - *Unsure about late rules later? The syllabus is your first stop before emailing.*
3. ✓ **Introduce yourself on the discussion board**
   - Post in the "Meet Your Classmates" forum: name, background, interests, one fun fact — and what you hope to build this semester.
   - *Example: "Hi everyone, I'm John Smith, a Data Science major. I recently built a chatbot for a class project and I'm excited to go deeper into Generative AI!"*
   - *This board is also where study groups form — a two-minute post can find you Sunday-review partners for the whole semester.*
4. ✓ **Set up your environment**
   - Install Python 3.9+ (Anaconda/Miniconda recommended) and Jupyter Notebook or VS Code — or confirm Google Colab access.
   - Follow the Week 1 setup guide on Canvas, run its verification notebook, and confirm PyTorch and Hugging Face libraries import cleanly.
   - *Test with a "Hello, World!" before touching PA1 — never debug your environment and your assignment at the same time.*

5. ✓ Map your semester

- Everything is already released (Aug 17) — skim all 8 PAs now and shortlist the 5 that excite you most.
- Add the deadlines to your personal calendar: **first milestone M1 is Tue, Sep 8; first assignment PA1 is Mon, Sep 14.**
- Start thinking about your project idea — approval is due within the first two weeks (by Mon, Aug 31), well before M1, so email me or book office hours early.

Pro tip: Students who complete these steps in the first 48 hours consistently report feeling more confident and less stressed all semester. Fifteen focused minutes now buys you months of calm.

❝ The 5-Minute Syllabus Scavenger Hunt (Optional — but Telling)

Think you know the syllabus? Five questions, five minutes, open-book. Every answer is in this document — and each one protects you from a December surprise. Jot your answers, then check the key below. (Optional fun: post your most surprising discovery in the "Meet Your Classmates" thread.)

1. Only one date all semester has two graded items due at once. What date is it, which two items collide — and how many free late days does each carry?

2. How many programming assignments must you complete, out of how many offered — and what happens to any extras you submit for grading?

3. Two deadlines in this course do not fall on a Monday. Which two, and what is the reason for each?

4. Which two milestones have the largest gap between them, how long is it, and what famous week sits inside that gap as built-in buffer?

5. What tag must appear on every programming assignment submission — and how is an untagged submission treated?

Answer key (no peeking until you've tried): 1) Mon, Oct 5 — PA4 and M2; each has its own 3 free late days. 2) Any 5 of 8; only your first five [FOR GRADING] submissions count, so choose deliberately. 3) M1 → Tue, Sep 8 (its Monday is Labor Day) and M3 → Wed, Oct 28 (placed off-Monday so it never shares a date with an assignment). 4) M3 → M4: 33 days, with Thanksgiving Break inside as buffer. 5) [FOR GRADING] or [FOR PRACTICE ONLY]; an untagged submission defaults to [FOR GRADING] and counts toward your five.

Scored 5/5? You know this course better than most students know theirs in week ten. Anything you missed — now you know exactly where to find it.