COURSE SYLLABUS

Generative AI

Course	MET CS 788: Generative AI
Modules	6
Topics Covered	Statistics, Feature Engineering, Neural Networks, Autoencoders & GANs, Transformers, Prompt Engineering & Agentic AI
Prerequisite	Python programing knowledge; familiarity with linear algebra and calculus recommended, and **MET CS 767**

Course Description

This course provides a comprehensive, hands-on journey through the core concepts and techniques that underpin modern machine learning and artificial intelligence. Beginning with the statistical and mathematical foundations essential for data understanding, the course progressively builds toward advanced neural network architectures, generative models, large language models, and the emerging field of agentic AI systems. Students will develop both theoretical knowledge and practical skills to design, train, evaluate, and deploy intelligent systems.

Required Book

Machine Learning and Artificial Intelligence: Concepts, Algorithms and Models, ISBN: 979-8992162103

Syllabus

Module	Title	Key Topics
1	Statistics & Probability for ML	Data types, descriptive statistics, probability theory, distributions, hypothesis testing, entropy, MLE
2	Feature Engineering	Feature selection & generation, normalization, scaling, transformation, categorical encoding, embeddings
3	Neural Network Fundamentals	Perceptron, MLP, activation functions, backpropagation, CNNs, RNNs, LSTM, regularization
4	Autoencoders & GANs	Latent space, encoder-decoder, sparse/denoising/VAE autoencoders, GAN architecture, training dynamics
5	Transformer Architecture	Seq2Seq, attention mechanisms, self-attention, multi-head attention, positional encoding, BERT
6	Prompt Engineering & Agentic AI	LLM fine-tuning (RLHF/DPO), MoE, prompt strategies, RAG, agentic patterns, multi-agent systems

Grading Criteria

40% of the grade comes from your assignments (a total of four assignments), and 60% of your grade is from the final exam. Final exam is closed book, and no cheat sheet or use of the Internet is allowed.
Following describes how your final grade is calculated:

A	95-100
A-	90-94.99
B+	85-89.99
B	80-84.99
B-	75-79.99
C+	70-74.99
C	65-69.99
F	<65

Please note that the challenge in this course is learning theoretical concepts and not coding. Therefore, be sure to prepare properly for the exam and read related sections in the book.

Module 1 Statistics & Probability for ML

Building the mathematical foundation for data-driven modeling

Module Overview

Statistics and probability are the backbone of machine learning. This module bridges descriptive and inferential statistics with the probabilistic reasoning needed to build, evaluate, and interpret ML models. Topics span data types and variable roles, central tendency and variability, probability theory, key distributions, hypothesis testing, information theory, and maximum likelihood estimation.

Learning Objectives

By the end of this module, students will be able to:

- Compute and interpret measures of central tendency (mean, median, mode) and variability (variance, standard deviation).
- Apply probability rules including union, intersection, conditional probability, and Bayes' theorem to ML problems.
- Describe and use common probability distributions: Normal, Uniform, Bernoulli, Binomial, Poisson, Beta, Weibull, and Power/Zipf.
- Interpret PMFs, PDFs, and CDFs and link them to real-world data modeling decisions.
- Perform and interpret parametric and non-parametric hypothesis tests and calculate effect sizes.
- Explain entropy, KL divergence, and cross-entropy and connect them to loss functions in ML.
- Apply Maximum Likelihood Estimation (MLE) and the Expectation-Maximization (EM) algorithm to parameter estimation tasks.

Module 2 Feature Engineering

Transforming raw data into machine-understandable representations

Module Overview

Feature engineering is the art and science of transforming raw data into the most informative representations for machine learning algorithms. This module covers the full pipeline: sanity checking, normalization, scaling, transformation, discretization, feature selection, feature generation, categorical encoding, and text/image embeddings.

Learning Objectives

By the end of this module, students will be able to:

- Explain the distinction between feature selection and feature generation, and when to apply each.
- Perform sanity checks on datasets and identify missing values, negative values, and format inconsistencies.
- Apply Min-Max, Z-score, L1, and L2 normalization/scaling to numerical data.
- Use logarithmic and Box-Cox transformations to reshape skewed distributions.
- Discretize and quantize continuous features into categorical bins.
- Encode categorical variables using one-hot encoding, ordinal encoding, and target encoding.
- Construct and interpret word embeddings (Word2Vec) and understand the concept of feature embeddings in deep learning.
- Select features using filter, wrapper, and embedded methods, and understand the trade-offs between bias and variance.

Module 3 Review on Neural Networks

From perceptrons to deep convolutional and recurrent architectures

Module Overview

This module provides a thorough grounding in artificial neural networks, from the single-neuron perceptron to multi-layer architectures capable of complex representation learning. It covers the mathematical mechanics of forward and backward propagation, key activation functions, convolutional neural networks (CNNs) for image data, and recurrent neural networks (RNNs/LSTMs/GRUs) for sequential data. Regularization and optimization strategies are also addressed.

Learning Objectives

By the end of this module, students will be able to:

- Describe the components of an artificial neuron (inputs, weights, bias, activation function) and formulate the neuron output equation.
- Explain the three-layer architecture of neural networks (input, hidden, output) and the role of depth in deep learning.
- Compare and apply activation functions: Step, Sigmoid, Tanh, ReLU, Leaky ReLU, ELU, Softmax, and GELU.
- Implement and trace the forward pass and backpropagation algorithm with gradient descent.
- Distinguish and apply optimization algorithms: SGD, Momentum, RMSProp, Adam.
- Build and interpret CNN architectures including convolution, pooling, and fully-connected layers; understand famous architectures (LeNet, VGG, ResNet).
- Explain the vanishing gradient problem and how LSTM and GRU architectures address it for sequential data.
- Apply regularization techniques: L1/L2, Dropout, Batch Normalization, and early stopping.

Module 4 Autoencoders & GANs

Self-supervised and generative models for representation learning

Module Overview

This module explores self-supervised and generative deep learning architectures. Students will learn how autoencoders compress data into a latent space and reconstruct it, enabling applications like denoising, anomaly detection, and representation learning. The module then introduces Generative Adversarial Networks (GANs), including their adversarial training dynamics, major variants, and practical considerations.

Learning Objectives

By the end of this module, students will be able to:

- ◻ Distinguish between generative and discriminative models and explain when each is appropriate.
- ◻ Describe the encoder-decoder architecture of autoencoders and explain the concept of latent space.
- ◻ Differentiate undercomplete from overcomplete autoencoders and explain the identity function problem.
- ◻ Explain and implement Sparse, Denoising, Contractive, and Stacked autoencoders.
- ◻ Describe Variational Autoencoders (VAEs), including the reparameterization trick and the ELBO loss.
- ◻ Explain the GAN framework: the generator-discriminator adversarial game and the minimax objective.
- ◻ Identify and describe key GAN variants: DCGAN, Conditional GAN, CycleGAN, StyleGAN.
- ◻ Recognize common GAN training challenges: mode collapse, vanishing gradients, and instability — and describe mitigation strategies.

Topics Covered

- **Generative vs. Discriminative Models:** Joint probability vs. conditional probability; stochastic vs. deterministic models; latent space intuition.
- **Autoencoder Architecture:** Encoder, decoder, latent representation; reconstruction loss; undercomplete vs. overcomplete autoencoders.
- **Autoencoder Training:** Backpropagation through encoder and decoder; cost function design; overfitting to identity.
- **Regularized Autoencoders:** Sparse Autoencoder (SAE); Denoising Autoencoder; Contractive Autoencoder; Stacked Autoencoders.
- **Variational Autoencoders (VAEs):** Probabilistic encoder; latent distribution (mean and variance); reparameterization trick; ELBO loss (reconstruction + KL term).
- **GAN Framework:** Generator and discriminator networks; minimax game; training procedure; Nash equilibrium.
- **GAN Variants:** DCGAN (deep convolutional); Conditional GAN; Wasserstein GAN (WGAN); CycleGAN; StyleGAN; Pix2Pix.
- **GAN Training Challenges:** Mode collapse; vanishing gradient in discriminator; training instability; evaluation metrics (FID, IS).

Module 5 Transformer Architecture

Attention mechanisms and the foundation of modern LLMs

Module Overview

The Transformer architecture has fundamentally reshaped natural language processing and beyond. This module traces the evolution from Seq2Seq RNN models through attention mechanisms to the full Transformer, culminating in BERT — the architecture that kicked off the modern era of large language models. Students gain a deep understanding of self-attention, multi-head attention, positional encoding, and how Transformers enable parallelized training on long sequences.

Learning Objectives

By the end of this module, students will be able to:

- ◻ Explain the Seq2Seq encoder-decoder architecture and its limitations with long sequences.
- ◻ Describe Bahdanau and Luong attention mechanisms and how they extend Seq2Seq models.
- ◻ Explain self-attention using the Query-Key-Value (QKV) framework and compute attention scores.
- ◻ Describe multi-head attention and explain how it captures diverse contextual relationships.
- ◻ Understand positional encoding and why it is necessary in Transformers.

- ☐ Trace the full Transformer encoder-decoder architecture, including feed-forward layers, layer normalization, and residual connections.

Topics Covered

- **Seq2Seq Models:** Encoder-decoder with RNNs/LSTMs; context vector; limitations for long sequences.
- **Attention Mechanisms:** Bahdanau (additive) attention; Luong (multiplicative) attention; attention weight visualization.
- **Self-Attention & QKV Framework:** Query, Key, Value vectors; scaled dot-product attention; attention score computation; softmax normalization.
- **Multi-Head Attention:** Parallel attention heads; concatenation and projection; capturing diverse relationships.
- **Transformer Architecture:** Encoder stack; decoder stack with masked self-attention and cross-attention; positional encoding; feed-forward sub-layers; layer normalization; residual connections.
- **Parallelization Advantages:** Eliminating sequential bottleneck of RNNs; training efficiency; long-range dependencies.

Module 6 Prompt Engineering & Agentic AI

From large language models to autonomous AI agents

Module Overview

The final module builds on the Transformer foundation to explore the cutting edge of AI: large language models, fine-tuning strategies, advanced prompting techniques, retrieval-augmented generation (RAG), and the rapidly evolving field of agentic AI systems. Students will understand how modern LLMs like GPT and ChatGPT/InstructGPT are built, how to engineer effective prompts, and how to design AI agents capable of planning, reasoning, and using tools autonomously.

Learning Objectives

By the end of this module, students will be able to:

- ☐ Trace the evolution from GPT-1 through GPT-3/ChatGPT (InstructGPT), including architectural and training dataset differences.
- ☐ Explain the Mixture of Experts (MoE) architecture and the role of sparse gating in scaling LLMs.
- ☐ Describe RLHF (Reinforcement Learning from Human Feedback) and DPO (Direct Preference Optimization) as LLM fine-tuning approaches.
- ☐ Apply zero-shot, few-shot, chain-of-thought, and self-consistency prompting strategies effectively.
- ☐ Implement Retrieval-Augmented Generation (RAG) to ground LLM responses in external knowledge.
- ☐ Design and evaluate agentic AI workflows using patterns such as ReAct, Reflection, Planning-and-Execute, and ReWOO.
- ☐ Architect multi-agent systems with role specialization, communication protocols, and shared memory.
- ☐ Identify key risks and safety considerations when deploying agentic AI systems in production.

Consolidated Learning Objectives

The following is a consolidated summary of all learning objectives across the six modules, suitable for course assessment planning and accreditation documentation.

Module 1 — Statistics & Probability for ML

- ☐ Classify and interpret data types for appropriate statistical treatment.
- ☐ Calculate and interpret descriptive statistics including mean, variance, covariance, and quartiles.

- ◻ Apply probability rules, conditional probability, and Bayes' theorem.
- ◻ Select and apply the correct probability distribution for a given data-generating process.
- ◻ Conduct hypothesis tests and interpret p-values, confidence intervals, and effect sizes.
- ◻ Connect entropy and KL divergence to ML loss function design.
- ◻ Apply MLE and EM to fit probabilistic models to data.

Module 2 — Feature Engineering

- ◻ Design a complete feature engineering pipeline for structured data.
- ◻ Apply scaling and normalization methods appropriate to the algorithm and data.
- ◻ Transform skewed distributions using log and Box-Cox transforms.
- ◻ Engineer categorical and text features including one-hot encoding and Word2Vec embeddings.
- ◻ Select features using filter, wrapper, and embedded methods.
- ◻ Generate new features through domain knowledge and mathematical operations.

Module 3 — Neural Network Fundamentals

- ◻ Formulate and compute the output of an artificial neuron and multi-layer network.
- ◻ Implement forward propagation and backpropagation for training neural networks.
- ◻ Choose and apply appropriate activation functions for specific tasks.
- ◻ Build CNN architectures for image processing tasks.
- ◻ Apply LSTM and GRU networks to sequential and time-series data.
- ◻ Apply regularization and optimization techniques to improve generalization.

Module 4 — Autoencoders & GANs

- ◻ Implement and train autoencoder architectures for dimensionality reduction and denoising.
- ◻ Apply sparse, denoising, and variational autoencoder variants to real-world tasks.
- ◻ Explain and implement the GAN adversarial training framework.
- ◻ Identify and mitigate common GAN failure modes.
- ◻ Evaluate generative models using FID and other metrics.

Module 5 — Transformer Architecture

- ◻ Implement scaled dot-product and multi-head self-attention.
- ◻ Explain how positional encoding enables the Transformer to model sequence order.
- ◻ Describe the full Transformer encoder-decoder architecture and its components.
- ◻ Fine-tune BERT-family models on downstream NLP tasks.
- ◻ Compare Transformer-based architectures for different NLP applications.

Module 6 — Prompt Engineering & Agentic AI

- ◻ Apply zero-shot, few-shot, and chain-of-thought prompting strategies.
- ◻ Implement RAG pipelines to augment LLM responses with retrieved knowledge.
- ◻ Design agentic workflows using ReAct, Reflection, and Planning-and-Execute patterns.
- ◻ Architect multi-agent systems with specialized roles and shared memory.
- ◻ Apply RLHF and DPO concepts to understand LLM alignment and fine-tuning.
- ◻ Evaluate the risks and safety implications of deploying autonomous AI agents.