MODULE 2 · STUDY NOTES

Introducing AI

What AI is, how machine learning learns, where deep learning and generative AI fit, and the AWS services that put them to work

Generative AI did not appear out of nowhere — it sits at the center of a set of nested fields that this module maps out. AI is the decades-old umbrella: making machines perform tasks that typically require human intelligence. Machine learning is the set of techniques at AI's core, training mathematical models on existing data so they can make predictions about new data. Deep learning is ML built from deeply layered neural networks, and generative AI is deep learning whose enormous pretrained models — foundation models — produce new content on request. Along the way, the module confronts the balancing act every AI project carries (powerful insights vs. bias, error, and opacity), lays out when ML is even the right tool, insists that good data is the fuel for all of it, and closes with the AWS services that deliver AI at every level of effort — from building your own models in SageMaker AI to calling a pre-trained service through a simple API.

LEARNING OBJECTIVES

- Define AI and machine learning (ML) terminology
- Explain the relationship between the concepts of AI, ML, deep learning, and generative AI
- Distinguish between scenarios that are suited for traditional software methods, traditional ML methods, or generative AI
- Identify the capabilities of AWS AI/ML services

2.1 AI overview

AI is an umbrella term — a field that has evolved since the 1950s, encompassing a variety of strategies and techniques for making machines perform tasks that typically require human intelligence. Familiar examples are really *specializations* under that umbrella: **computer vision** (a computer's ability to “see” things in images, such as facial recognition) and **smart devices** such as Siri and Alexa. At the center of AI sits a broad set of techniques called **machine learning (ML)** — and the specializations actually incorporate ML techniques rather than compete with them. Generative AI is also a type of machine learning. Because ML is so central, you will often see the terms interchanged or combined as *AI/ML*.

BENEFITS – WHY ORGANIZATIONS ADOPT AI	RISKS – WHAT MUST BE MANAGED
insights and efficiency beyond traditional programming	*the flip side of imitating human thinking*
▪ Analyzes vast amounts of complex data to derive insights that would be impossible with traditional programming methods ▪ Automates repetitive or tedious tasks and increases efficiency ▪ Frees human resources to focus on strategic and creative work	▪ Biases and errors in the underlying data and algorithms can lead to unfair or inaccurate decision-making ▪ AI logic is less transparent than traditional systems — judging whether there are problems is harder

Both sides of that ledger are real. On the promise side, AI can perform complex analysis across massive amounts of data — including sounds, images, and environmental information — letting organizations identify and respond to issues that would otherwise demand far larger numbers of human experts. Organizations are applying AI to global problems including **wildlife protection, food insecurity, and disaster recovery**. On the challenge side, studies have reported that some algorithms used in health care demonstrate **racial bias** when recommending treatment approaches — patients might not get the care they need — and lawyers have submitted court documents containing **fake citations** produced by AI tools, drawing lawsuits. A useful rule of thumb: as an application's complexity grows, the potential risks tend to grow with it, and explaining what the system is doing gets harder.

TABLE 2.1 The three pillars of responsible AI practice

PILLAR	WHAT IT DEMANDS
Accurate, fair, safe, and secure	Keep everyone and their data safe; make sure output is valid and fair
Explainable and transparent	Provide mechanisms to validate and audit what happens behind the scenes; empower stakeholders to make informed choices about their interactions with the system
Governed and controlled	Incorporate responsible best practices across the full scope of the AI system; provide mechanisms to steer system behavior

Responsible AI encompasses the best practices, guidelines, and tools for ensuring that any AI solution's outputs are protected against the risks inherent in systems that imitate human thinking. It is a rapidly evolving field — a later module, *Practicing Generative AI Responsibly*, examines responsible AI specifically for generative solutions.

> **COMMON PITFALL**
>
> Responsible AI is not just the builder's problem, and not just a generative AI problem. Regardless of the type of AI system you **use or build**, keeping it responsible is always a top priority — and as ethical users and practitioners, it is everyone's job to continuously question whether an AI application properly protects its users and provides fair outcomes.

2.2 Machine learning concepts

ML is a group of AI techniques that use existing data to train a mathematical model. The model learns to recognize patterns in the training data so that it can make an accurate prediction when faced with new input data. The **model is the end result of the training process** — the algorithm has been applied to a dataset and has learned to make predictions or decisions. Algorithms have evolved into more advanced model architectures such as **neural networks**, defined by complex arrangements of layers, neurons, and connections; those advances, combined with vast amounts of available data and more powerful cloud compute, opened up new ML techniques — including generative AI. A “prediction” takes many shapes: answering a question, categorizing something, or making a decision — from something specific like *is this transaction fraudulent?* to something broad like a restaurant recommendation.

> **WORKED EXAMPLE – A MODEL IS LIKE A BRAIN**
>
> How does a young child learn to tell dogs from cats? Not from a list of criteria that make a dog a dog. You point at dogs and cats in picture books and in real life and say the words; the child's brain builds a mental model from those examples, and the more animals they have seen, the better they distinguish them. That mental model is a combination of the data the child was exposed to and their own internal mechanisms for processing and storing it. Machine learning works the same way: instead of coding rules for something too complex to define as rules, a data scientist selects **training data** and an appropriate **algorithm** and exposes the model to lots of examples. The data you have and the problem you are solving dictate those choices — and training includes evaluating how well the model performs, then tuning, collecting more data, and running further train-and-test cycles until the results meet the goal.

Trained models make inferences. Once a model is trained, tuned, and tested, you deploy it to an environment where it can predict on live data. A credit card company's fraud model, for example, gets integrated into the production transaction-processing application: each incoming transaction is sent to the model, and its output determines the next processing step. Each individual prediction on new data is an **inference** — the model applying its learned knowledge to new input to produce a new output. Deployment is not the finish line: inferences must be **monitored and evaluated** in production. Poor performance can point to problems in the training data or process, and even a model that starts strong can lose accuracy as the world changes — fraudsters develop new techniques that follow different patterns than the model learned. Models might need to be **retrained** to learn new things.

CHOOSE TRADITIONAL DATA ANALYTICS WHEN…	CONSIDER ML ONLY IF ANALYTICS FALLS SHORT – THEN CHECK:
traditional programming can carry the load ▪ Business rules can be hardcoded ▪ Variables are limited ▪ Statistical modeling can solve the problem ▪ Analytics provides answers quickly enough	*four gates before you build* ▪ Enough relevant, high-quality training data exists (or can be obtained) ▪ ML can deliver the required accuracy, transparency, and speed ▪ The organization has the skills and resources to build **and sustain** the solution ▪ The cost of ML is a good match to the impact of the problem

> **WORKED EXAMPLE – ONE RETAILER, TWO PROBLEMS, TWO APPROACHES**
>
> A retail company managing **inventory levels** analyzes historical sales data each month and forecasts demand with straightforward statistical methods. Rules are simple, variables are limited, and there is no real-time requirement — **traditional data analytics** is the better choice. The same company wants **personalized marketing**: it collects purchase history, browsing behavior, demographics, and social media interactions, segments customers into groups by buying patterns, and suggests products of interest **in real time**. That complexity makes it a good candidate for **ML** — provided the four gates (data, requirements, skills, cost vs. impact) all pass. Even a problem complex enough for ML deserves that check: an ML model's predictions might not be precise enough, might not satisfy a requirement for transparent explanations of decisions, or might not produce output fast enough. Cloud resources lower the cost of experimenting toward a cost-optimized approach.

GOOD DATA IS CRITICAL TO HIGH-VALUE, RESPONSIBLE OUTCOMES

- **Volume** — is there enough data to train effectively? The more critical the prediction, the more essential sufficient high-quality data becomes.
- **Quality** — grainy photos vs. clear examples: poor inputs teach poor lessons.
- **Bias and representation** — over- or under-represented categories skew the model; a child who has only seen Labrador retrievers may not recognize a Chihuahua.
- **Timeliness** — dogs and cats don't change, but product-recommendation data must reflect very recent trends.
- **Managing the data** — protect any personally identifiable information (PII), secure the data through development and deployment, and apply governance to avoid duplication and inconsistencies.
- Data preparation and pre-processing typically consume a **very large percentage of an ML project's timeline**.

TABLE 2.2 General data types that ML might use

TYPE	STRUCTURE	EXAMPLES	WORKING WITH IT
Structured	Rows and columns; well-defined schema standardizing elements and relations	Relational database tables, spreadsheets	Simpler to query, but not very flexible
Semistructured	Elements and attributes; self-describing tags or markers	CSV, JSON, XML; email fields	Less rigid than structured, still organized enough to work with
Unstructured	No predefined structure	Images, movies, audio, text documents, clickstream data	More complex to analyze; with the right tools, the most flexible

Labeled data identifies the target of the prediction. In the dog-vs-cat analogy the data was labeled — each image was identified as *dog* or *cat*. Labeling can be that explicit (a subject matter expert tagging medical images that contain a marker for a condition), or it can be a **field in the training data** — a fraud-detection dataset of historical transactions with a column indicating whether each was fraud. With labeled data, the model uses that information to find common patterns and learns to identify the labeled element in new data. With **unlabeled** data, the approach and model must find the relevant patterns on their own — which generally requires **more training data and more complex models**. Sometimes labeling is itself a preprocessing step, done by subject matter experts or even by another ML technique.

TABLE 2.3 Basic ML paradigms and common problem types

PARADIGM	TRAINING DATA	EXAMPLE PROBLEM TYPES	APPLICATION
Supervised	Labeled	Classification, regression	Decide whether a patient has a disease based on diagnostic test results
Unsupervised	Unlabeled	Clustering, anomaly detection	Analyze spending patterns to identify possible fraud
Reinforcement	Rewards or penalties derived from the environment	Robotics, inventory control	Teach a robot to navigate around a room

These paradigms are used and combined in a variety of ways — different approaches might serve different phases of the same project. This vocabulary is what you need to describe deep learning in the next section.

> **COMMON PITFALL**
>
> Don't pick the technology first. The recurring theme of this module is choosing the right solution for the **business problem** — not choosing ML (or generative AI) and then figuring out how to make it fit. If analytics answers the question, use analytics.

2.3 Machine learning, deep learning, and generative AI

Neural networks consist of layers: an **input layer** where data is fed in, one or more **hidden layers** where the complex processing occurs, and an **output layer** that produces the final prediction. More complex models have more layers. What distinguishes **deep learning** from other ML is the **volume and variety of data** the models train on and the resulting **depth** (number of layers) of the model itself. These deeply layered models excel at finding patterns in **unstructured data** and modeling complex relationships between inputs and outputs — exactly what complex AI tasks demand.

The contrast with traditional ML is scope. A traditional ML model has a well-defined dataset and is trained to use a set of data points to make a **specific prediction —** trained on historical transactions labeled fraud/not-fraud, it predicts whether a new transaction is fraud. A deep learning model learns very complex patterns in large, varied, unstructured datasets and is designed to solve a problem or perform a set of **interrelated, complex tasks within a specific domain**. A self-driving car's model ingests information about its environment, navigational information, and traffic rules to make decisions about operating the car — a singular goal composed of many individual predictions. Other deep learning use cases: weather prediction and personalized recommendations.

Generative AI uses deep learning to create new content. Its models are very large deep learning models called **foundation models (FMs)**, pretrained on massive amounts of data. That pretraining lets them respond to **generalized requests — regardless of whether the specific request was part of their training**. Their predictions actually produce *new content*: natural-language responses in a conversational dialog, or a novel image generated from a requested description. This is why FMs appeal to a much broader set of use cases — the trained model can serve applications that **haven't been defined in advance**. Example use cases: conversational chat-based applications, coding assistants, and image generators. The FM concept is at the heart of generative AI, and the next module expands on it.

TABLE 2.4 Traditional ML vs. deep learning vs. generative AI

DIMENSION	TRADITIONAL ML	DEEP LEARNING	GENERATIVE AI
Model	Algorithm trained into a model	Deeply layered neural network	Foundation model (very large deep learning model)
Training data	A specific, well-defined dataset	Very large amounts of varied, unstructured data	Massive pretraining data
Output	One specific prediction	Complex, interrelated tasks in one domain	New content; generalized responses across broad domains
Example	Fraud yes/no on a transaction	Self-driving cars, weather prediction, recommendations	Chat applications, coding assistants, image generators

WORKED EXAMPLE – THE MODULE'S SAMPLE EXAM QUESTION

A customer service department wants to implement a system that can automatically categorize incoming support tickets. Which approach would be MOST suitable? Isolate the key words first: **automatically**, **categorize**, **incoming support tickets**. Categorizing is classification — a classic supervised-learning problem — so **traditional ML methods** are the answer: they learn from historical ticket data to accurately categorize new tickets based on content. Traditional software methods would be too rigid to adapt to the variety of incoming tickets; generative AI has many applications but is **not the most cost-effective** choice for categorizing existing content; and manual sorting would be time-consuming, inconsistent, and inefficient compared to an automated ML approach.

COMMON PITFALL

Deep learning ≠ generalized. Deep learning models are **designed for a specific domain** — the self-driving car's model drives cars. The generalized, any-request behavior belongs to **foundation models**, which train on vast unstructured data and provide generalized outputs. On scenario questions, match the scope of the task to the scope of the approach.

2.4 AWS AI services

AWS services span a spectrum of control vs. convenience, giving an organization flexibility about **where to draw the line between control of infrastructure and freedom to focus on the business side** of a solution. At the most hands-on end, teams run the **full ML lifecycle —** collecting and cleaning data, then training, testing, and deploying a model. For that path, AWS offers **Amazon EC2 instances designed specifically to optimize training performance**, and separate instance types designed for the distinct demands of **delivering inferences in production**. **Amazon SageMaker AI** provides an integrated environment for building, training, and deploying models, including managed infrastructure tools and workflows; later modules revisit SageMaker AI in generative AI contexts.

For teams that want to start **without investing in infrastructure and without training their own models**, **Amazon Bedrock** provides access to popular **foundation models through an API —** regardless of a model's underlying complexity, a business reaches its capabilities through a simple, well-defined interface. Bedrock is designed for building scalable generative AI applications. One step further, AWS offers a group of **pre-trained AI services** that let you integrate AI into applications **without dealing with models at all**: **Amazon Q** is a generative AI tool covered in upcoming modules, while purpose-built services handle specific AI tasks **—** **Amazon Transcribe** creates text output from audio files, and **Amazon Comprehend** derives insights from text documents. These lists are not exhaustive; the AWS website's Artificial Intelligence section catalogs all AI services and AI-focused EC2 instance types.

TABLE 2.5 Key AWS AI/ML services and where they fit

SERVICE	LAYER	CAPABILITY
EC2 AI instances	Infrastructure	Compute purpose-built for training workloads and for production inference workloads
Amazon SageMaker AI	Full lifecycle	Integrated environment to build, train, and deploy models with managed infrastructure tools and workflows
Amazon Bedrock	Foundation models	API access to popular FMs for building scalable generative AI applications
Amazon Q	Pre-trained service	Generative AI assistant (detailed in later modules)
Amazon Transcribe	Pre-trained service	Creates text outputs from audio files
Amazon Translate	Pre-trained service	Translates text between languages
Amazon Polly	Pre-trained service	Generates synthesized speech from text
Amazon Comprehend	Pre-trained service	Derives insights from text documents

WORKED EXAMPLE – THE MODULE DEMO: ENGLISH AUDIO TO SPANISH SPEECH

Pre-trained services compose into pipelines. The module's demo translates an audio file from English to Spanish using three services in sequence: **Amazon Transcribe** develops a transcript from the audio → **Amazon Translate** translates the English text into Spanish → **Amazon Polly** generates synthesized speech from the translated file. No model is trained, tuned, or hosted anywhere in the pipeline — each service is consumed as-is.

COMMON PITFALL

SageMaker AI and Bedrock are not interchangeable. **SageMaker AI** is the integrated environment for building, training, and deploying *your own* models across the full lifecycle; **Bedrock** is API access to *existing* foundation models for generative AI applications, with no infrastructure to manage. “Which service?” questions turn on whether the scenario involves training a model or calling one.

Module summary

- AI is the umbrella field (evolving since the 1950s) of making machines perform tasks that typically require human intelligence; computer vision and smart devices are AI specializations that incorporate ML — the technique set at AI's center.
- ML trains a mathematical model on existing data to recognize patterns; the model's prediction on new input is an inference. Deployed models must be monitored and retrained as real-world patterns change.
- AI's benefits (insights impossible with traditional programming, automation, efficiency) must be balanced against its risks (bias, error, opacity, weak accountability) through responsible AI: accurate/fair/safe/secure, explainable/transparent, governed/controlled.
- Choose traditional data analytics when rules can be hardcoded and variables are limited; choose ML only when data, accuracy/transparency/speed requirements, skills, and cost-vs-impact all check out.

- Good data decides outcomes: volume, quality, bias/representation, and timeliness — plus PII protection, security, and governance. Data comes structured, semistructured, or unstructured; labels identify the prediction target; unlabeled data demands more data and more complex models.
- The three ML paradigms: supervised (labeled → classification/regression), unsupervised (unlabeled → clustering/anomaly detection), reinforcement (environment rewards/penalties → robotics, inventory control).
- Deep learning is ML on deeply layered neural networks (input → hidden → output layers) trained on large varied data — strong on unstructured data and complex relationships, but designed for a specific domain. Generative AI is deep learning whose foundation models generate new content for generalized requests.
- AWS covers every level of effort: purpose-built EC2 instances and SageMaker AI for the full lifecycle, Bedrock for foundation models via API, and pre-trained services — Amazon Q, Transcribe, Translate, Polly, Comprehend — to plug AI straight into applications.

Review questions

1. Explain the relationship between AI, ML, deep learning, and generative AI in one sentence each.

Answer: AI is the broad field of making machines do human-intelligence tasks; ML is the set of AI techniques that train models from data to make predictions; deep learning is ML using deeply layered neural networks trained on very large data; generative AI is deep learning that uses pretrained foundation models to generate new content.

2. What is the difference between training and inference?

Answer: Training applies an algorithm to existing data so the model learns patterns; inference is the trained model applying that learning to new input data to produce an output in deployment.

3. A fraud model performed well at launch but its accuracy is slipping. Why can that happen, and what is the remedy?

Answer: Real-world patterns changed — fraudsters developed new techniques that don't match the patterns the model learned. Monitor production inferences and retrain the model on data reflecting the new patterns.

4. A retailer needs (a) monthly inventory forecasts from historical sales and (b) real-time personalized product recommendations. Which approach fits each, and why?

Answer: (a) Traditional data analytics — limited variables, hardcodable rules, statistical forecasting, no real-time need. (b) ML — many variables (purchase history, browsing, demographics, social), customer segmentation, real-time suggestions.

5. Name the four data-quality considerations for choosing training data, and the three data-management concerns.

Answer: Quality: volume (enough data), quality, bias/representation, timeliness. Management: protecting PII, securing data through development and deployment, and governance to avoid duplication and inconsistencies.

6. What makes a foundation model different from a traditional ML model?

Answer: A traditional ML model trains on a specific dataset to make one specific prediction; a foundation model is a very large deep learning model pretrained on massive data that answers generalized requests — even ones outside its training — and produces new content, serving applications not defined in advance.

7. **List the three pillars of responsible AI and what each requires.**

   Answer: Accurate, fair, safe, and secure — protect people and data, ensure valid and fair output. Explainable and transparent — mechanisms to validate and audit behavior so stakeholders can make informed choices. Governed and controlled — responsible practices across the system's full scope, with mechanisms to steer behavior.

8. **Your team wants AI capabilities but will not train models or manage infrastructure. What are the AWS options?**

   Answer: Amazon Bedrock for foundation-model access through an API (generative AI apps), or the pre-trained AI services — Amazon Q, Transcribe, Translate, Polly, Comprehend — which integrate AI with no model work at all, and can be chained together (e.g., Transcribe → Translate → Polly).