Content outline

This exam guide includes weightings, content domains, and task statements for the exam. This guide does not provide a comprehensive list of the content on the exam. However, additional context for each task statement is available to help you prepare for the exam.

The exam has the following content domains and weightings:

• Domain 1: Fundamentals of AI and ML (20% of scored content)

• Domain 2: Fundamentals of Generative AI (24% of scored content)

• Domain 3: Applications of Foundation Models (28% of scored content)

• Domain 4: Guidelines for Responsible AI (14% of scored content)

• Domain 5: Security, Compliance, and Governance for AI Solutions (14% of scored content)

Domain 1: Fundamentals of AI and ML

Task Statement 1.1: Explain basic AI concepts and terminologies.

Objectives:

• Define basic AI terms (for example, AI, ML, deep learning, neural networks, computer vision, natural language processing [NLP], model, algorithm, training and inferencing, bias, fairness, fit, large language model [LLM]).

• Describe the similarities and differences between AI, ML, and deep learning.

• Describe various types of inferencing (for example, batch, real-time).

• Describe the different types of data in AI models (for example, labeled and unlabeled, tabular, time-series, image, text, structured and unstructured).

• Describe supervised learning, unsupervised learning, and reinforcement learning.

Task Statement 1.2: Identify practical use cases for AI. Objectives:

• Recognize applications where AI/ML can provide value (for example, assist human decision making, solution scalability, automation).

• Determine when AI/ML solutions are not appropriate (for example, costbenefit analyses, situations when a specific outcome is needed instead of a prediction).

• Select the appropriate ML techniques for specific use cases (for example, regression, classification, clustering).

• Identify examples of real-world AI applications (for example, computer vision, NLP, speech recognition, recommendation systems, fraud detection, forecasting).

• Explain the capabilities of AWS managed AI/ML services (for example, SageMaker, Amazon Transcribe, Amazon Translate, Amazon Comprehend, Amazon Lex, Amazon Polly).

Task Statement 1.3: Describe the ML development lifecycle. Objectives:

• Describe components of an ML pipeline (for example, data collection, exploratory data analysis [EDA], data pre-processing, feature engineering, model training, hyperparameter tuning, evaluation, deployment, monitoring).

• Understand sources of ML models (for example, open source pre-trained models, training custom models).

• Describe methods to use a model in production (for example, managed API service, self-hosted API).

• Identify relevant AWS services and features for each stage of an ML pipeline (for example, SageMaker, Amazon SageMaker Data Wrangler, Amazon SageMaker Feature Store, Amazon SageMaker Model Monitor).

• Understand fundamental concepts of ML operations (MLOps) (for example, experimentation, repeatable processes, scalable systems, managing technical debt, achieving production readiness, model monitoring, model re-training).

• Understand model performance metrics (for example, accuracy, Area Under the ROC Curve [AUC], F1 score) and business metrics (for example, cost per user, development costs, customer feedback, return on investment [ROI]) to evaluate ML models.

Domain 2: Fundamentals of Generative AI

Task Statement 2.1: Explain the basic concepts of generative AI.

Objectives:

• Understand foundational generative AI concepts (for example, tokens, chunking, embeddings, vectors, prompt engineering, transformer-based LLMs, foundation models, multi-modal models, diffusion models).

• Identify potential use cases for generative AI models (for example, image, video, and audio generation; summarization; chatbots; translation; code generation; customer service agents; search; recommendation engines).

• Describe the foundation model lifecycle (for example, data selection, model selection, pre-training, fine-tuning, evaluation, deployment, feedback).

Task Statement 2.2: Understand the capabilities and limitations of generative AI for solving business problems.

Objectives:

• Describe the advantages of generative AI (for example, adaptability, responsiveness, simplicity).

• Identify disadvantages of generative AI solutions (for example, hallucinations, interpretability, inaccuracy, nondeterminism).

• Understand various factors to select appropriate generative AI models (for example, model types, performance requirements, capabilities, constraints, compliance).

• Determine business value and metrics for generative AI applications (for example, cross-domain performance, efficiency, conversion rate, average revenue per user, accuracy, customer lifetime value).

Task Statement 2.3: Describe AWS infrastructure and technologies for building generative AI applications.

Objectives:

• Identify AWS services and features to develop generative AI applications (for example, Amazon SageMaker JumpStart; Amazon Bedrock; PartyRock, an Amazon Bedrock Playground; Amazon Q).

• Describe the advantages of using AWS generative AI services to build applications (for example, accessibility, lower barrier to entry, efficiency, cost-effectiveness, speed to market, ability to meet business objectives).

• Understand the benefits of AWS infrastructure for generative AI applications (for example, security, compliance, responsibility, safety).

• Understand cost tradeoffs of AWS generative AI services (for example, responsiveness, availability, redundancy, performance, regional coverage, token-based pricing, provision throughput, custom models).

Domain 3: Applications of Foundation Models

Task Statement 3.1: Describe design considerations for applications that use foundation models.

Objectives:

• Identify selection criteria to choose pre-trained models (for example, cost, modality, latency, multi-lingual, model size, model complexity, customization, input/output length).

• Understand the effect of inference parameters on model responses (for example, temperature, input/output length).

• Define Retrieval Augmented Generation (RAG) and describe its business applications (for example, Amazon Bedrock, knowledge base).

• Identify AWS services that help store embeddings within vector databases (for example, Amazon OpenSearch Service, Amazon Aurora, Amazon Neptune, Amazon DocumentDB [with MongoDB compatibility], Amazon RDS for PostgreSQL).

• Explain the cost tradeoffs of various approaches to foundation model customization (for example, pre-training, fine-tuning, in-context learning, RAG).

• Understand the role of agents in multi-step tasks (for example, Agents for Amazon Bedrock).

Task Statement 3.2: Choose effective prompt engineering techniques.

Objectives:

• Describe the concepts and constructs of prompt engineering (for example, context, instruction, negative prompts, model latent space).

• Understand techniques for prompt engineering (for example, chain-ofthought, zero-shot, single-shot, few-shot, prompt templates).

• Understand the benefits and best practices for prompt engineering (for example, response quality improvement, experimentation, guardrails, discovery, specificity and concision, using multiple comments).

• Define potential risks and limitations of prompt engineering (for example, exposure, poisoning, hijacking, jailbreaking).

Task Statement 3.3: Describe the training and fine-tuning process for foundation models.

Objectives:

• Describe the key elements of training a foundation model (for example, pre-training, fine-tuning, continuous pre-training).

• Define methods for fine-tuning a foundation model (for example, instruction tuning, adapting models for specific domains, transfer learning, continuous pre-training).

• Describe how to prepare data to fine-tune a foundation model (for example, data curation, governance, size, labeling, representativeness, reinforcement learning from human feedback [RLHF]).

Task Statement 3.4: Describe methods to evaluate foundation model performance.

Objectives:

• Understand approaches to evaluate foundation model performance (for example, human evaluation, benchmark datasets).

• Identify relevant metrics to assess foundation model performance (for example, Recall-Oriented Understudy for Gisting Evaluation [ROUGE], Bilingual Evaluation Understudy [BLEU], BERTScore).

• Determine whether a foundation model effectively meets business objectives (for example, productivity, user engagement, task engineering).

Domain 4: Guidelines for Responsible AI

Task Statement 4.1: Explain the development of AI systems that are responsible.

Objectives:

• Identify features of responsible AI (for example, bias, fairness, inclusivity, robustness, safety, veracity).

• Understand how to use tools to identify features of responsible AI (for example, Guardrails for Amazon Bedrock).

• Understand responsible practices to select a model (for example, environmental considerations, sustainability).

• Identify legal risks of working with generative AI (for example, intellectual property infringement claims, biased model outputs, loss of customer trust, end user risk, hallucinations).

• Identify characteristics of datasets (for example, inclusivity, diversity, curated data sources, balanced datasets).

• Understand effects of bias and variance (for example, effects on demographic groups, inaccuracy, overfitting, underfitting).

• Describe tools to detect and monitor bias, trustworthiness, and truthfulness (for example, analyzing label quality, human audits, subgroup analysis, Amazon SageMaker Clarify, SageMaker Model Monitor, Amazon Augmented AI [Amazon A2I]).

Task Statement 4.2: Recognize the importance of transparent and explainable models.

Objectives:

• Understand the differences between models that are transparent and explainable and models that are not transparent and explainable.

• Understand the tools to identify transparent and explainable models (for example, Amazon SageMaker Model Cards, open source models, data, licensing).

• Identify tradeoffs between model safety and transparency (for example, measure interpretability and performance).

• Understand principles of human-centered design for explainable AI.

Domain 5: Security, Compliance, and Governance for AI Solutions

Task Statement 5.1: Explain methods to secure AI systems.

Objectives:

• Identify AWS services and features to secure AI systems (for example, IAM roles, policies, and permissions; encryption; Amazon Macie; AWS PrivateLink; AWS shared responsibility model).

• Understand the concept of source citation and documenting data origins (for example, data lineage, data cataloging, SageMaker Model Cards).

• Describe best practices for secure data engineering (for example, assessing data quality, implementing privacy-enhancing technologies, data access control, data integrity).

• Understand security and privacy considerations for AI systems (for example, application security, threat detection, vulnerability management, infrastructure protection, prompt injection, encryption at rest and in transit).

Task Statement 5.2: Recognize governance and compliance regulations for AI systems.

Objectives:

• Identify regulatory compliance standards for AI systems (for example, International Organization for Standardization [ISO], System and Organization Controls [SOC], algorithm accountability laws).

• Identify AWS services and features to assist with governance and regulation compliance (for example, AWS Config, Amazon Inspector, AWS Audit Manager, AWS Artifact, AWS CloudTrail, AWS Trusted Advisor).

• Describe data governance strategies (for example, data lifecycles, logging, residency, monitoring, observation, retention).

• Describe processes to follow governance protocols (for example, policies, review cadence, review strategies, governance frameworks such as the Generative AI Security Scoping Matrix, transparency standards, team training requirements).