Day 1 - Recap

Module 1 - GAI, LLM, Vector + Multimodal Offerings

A. Enterprise Model Evaluation Framework
① Capability / Quality
② Performance
③ Cost/Risk

B. Mitigating Strategies + mitigating limitations
✓ Context gaps → RAG
✓ Knowledge cutoff → RAG
✓ Context limits → chunking + summarization
✓ Hallucinations → fact-checking + Confidence scoring

AWS Service Integration
↳ Bedrock + KB, Guardrails, Agents

C. Dynamic Model Selection Architecture Complexity

S → [Intent Classifier] → [Complexity Scorer] → [Model Selector] → [Bedrock, SM, ...]

↓

[Output Validator]

③ Resilience + Cost
① Circuit Breaker → To stop calling the failing Provider
② Cost Guard
③ Multi-region failover

Module 2: → Feature Stores Models → Input Quality
↳ Data Data Processing for FMs

① Data Validation + QA

Completeness → Accuracy → Consistency → Timeliness
(Missing Data) (Value Validity Verification) (Schema format) (Freshness)

AWS Glue Data Quality → Automated Rule-Based scoring
↓
Lambda Validations → Domain-specific logic

② Multi-Model Pipeline

Processing Patterns:
① Parallel ✓
② Sequential ✓
③ Hybrid ✓
→ AWS Service
Step Functions

③ Input Optimization + Performance
① Token efficiency
② for chat apps → Sliding window + summarize old messages
③ for throughput → Batching + caching (SQS Lambda) + caching (ElastiCache) + S3

Day 2 - Agenda:

① Vector Databases + Retrieval Augmentation
Demo: Advanced Chunking Strategies ①

② Prompt Engineering & Governance.
- Adv. RE framework
- Complex prompt systems
- Enterprise prompt governance & management
Demo: prompt decomposition ②
prompt evaluation ③
Amazon Bedrock flows ④

Lab: Develop cross-dim patterns with Bedrock Flows → SageMaker Notebook

③ Implement Agentic AI frameworks with Amazon Bedrock Agentcore
Langchain/langgraph → open source framework → Document

AWS → Strands ① Frameworks
② Amazon Bedrock Agentcore

Demo: → ③

④ AI Safety & Security
✓ Comprehensive content safety implementation
✓ Privacy-preserving AI
✓ AI governance & compliance frameworks

FM | LLM → Pretrained Models (Internet scale Data)
↳ Raw + Multi-lingual

Transformer → Input (Tokens)
↳ ① Tokenization & Embedding

I am learning AWS generative AI

Tokens → numerical representation → [ ]

KB (Bedrock) → vector DB → AWS

Data Source → Parsing (Text) → Embedding (Token → num text → vectors) → Vector DB

Precision vs Recall

Precision:
→ predicted positives correct

TP — correct
FP — wrong

Precision = TP / (TP+FP) = 8/(8+2) = 80%

Recall:
→ actual positives correctly identified

Recall = TP / (TP+FN) = 8/(8+4) = 8/12 = 0.67 → 67%

TP — correct
FN — missed

FP & FN IMP
↳ High Precision or Recall

Amazon Bedrock AgentCore

① AgentCore Runtime → Deploy & scale Dynamic AI Agents & tools → Open Source framework, Bedrock & Nonbedrock

② AgentCore Memory
i) SHORT-TERM → multi-turn conversation context-aware
ii) LONG-TERM → persist across sessions

③ AgentCore Gateway + AgentCore Identity

AgentCore Identity Architecture
↳ Cognito

AgentCore ⇒ AWS Regions
US-East-1
us-west-1

Agent Runtime
Agent Runtime
Agent Runtime