TABLE OF CONTENTS

SECTION 1: INTRODUCTION TO AI

What is AI? AI is a field of computer science to solve problems we associate with human intelligence, e.g. image creation, image recognition, etc.

Training dataset → Train AI Model → Use Model → New Input → New Classification

History:

- Alan Turing: Turing Test
- John McArthy: AI
- MYCIN test
- ML and Data Mining
- Deep Blue - Chess
- 2010: Deep Learning—Googles AlphaGo

IDP: Intelligent Document Processing

SECTION 3: INTRODUCTION TO AWS CLOUD

Website: Client → Network (Router →Switch) → Server
Client IP → Server IP

Server: CPU (Compute)
RAM → Brain
Storage: File, Database
Network

Cloud Computing: On-demand delivery of compute power, database storage and applications

Private: Not exposed to the public
Public: Amazon, Google, Microsoft—delivered over the internet
Hybrid: Part private, part public

5 Characteristics:

- On demand
- Broad network access
- Multi-tenancy: shared infra-structure
- Rapid elasticity and scalability
- Measured service

6 Advantages:

- Trade capex for opex
- Economies of scale
- No guessing of capacity
- Increase speed and agility
- No money on data centers nor maintenance

IaaS: raw network, compute and storage: e.g. EC2
Paas: remove need for managing underlying infrastructure. E,g Beanstalk
Saas: completed product—everything managed by AWS, e.g. Rekognition

AWS Is Global: Region → Availability Zone → Data Center

Region: A cluster of data centers
Compliance, Proximity, Services (not all regions have all services), pricing

AZ: At least 3 per region
1 or more discrete data centers
Separate from each other, connected by high-speed network

Edge Locations: Content delivered to end user

AWS Services:

Global: IAM, Route 53, CloudFront, WAF
Region: EC2, Lamda, Beanstalk, Rekognition

Shared Responsibility: “Security In The Cloud vs Security Of The Cloud”

SECTION 5: AMAZON BEDROCK AND GEN AI

Overview Of GEN AI

One of the main topics on the exam

What is Gen AI? Gen AI→ Deep Learning→ML
Used to generate new data from trained on data

Unlabeled data→Train A Foundation Model→Tasks

Foundation Models: Generate text, images, summarize text, chatbot

- Foundation models use self-supervised learning to create labels from input data. This means no one has instructed or trained the model with labeled training data sets.
- $10M to train a foundation model
- E.g., GPT-4o, OpenAI, Meta, Amazon, Google, Anthropic
- Open source vs License: BERT vs OpenAI, Anthropic

Large Language Model: Generate coherent human-like text

- Chat-GPT
- Trained on large amounts of data
- Books, websites, articles, etc.
- Translation, summarization, etc.
- **Interaction**: Prompt→Content→New Content
- **Non-deterministic**: may get different answers

Generative Language Models: Generate a list of words alongside probabilities.

Generative AI For Images:

- Images from prompt
- Images from image
- Text from images
- How do they work? **Diffusion Model (Stable Diffusion)**
    - Images→Noise→Noise→Noise→Noise (Forward Diffusion)
    - Repeat for a lot of pictures
    - Noise→De-Noise→Cat With Computer (Reverse diffusion)

Amazon Bedrock - Build Generative AI Applications

Characteristics

- Build Generative AI

- Fully Manager Service
- Keep control of all data used to train the model
- Pay-per-use pricing model
- Unified APIs
- Leverage a large array of Foundation Models
- RAG (Knowledge Basis), LLM, Agents

Foundation Models:

- AI21Labs, Cohere, Meta, Stability.ai, Meta
- Select model→Make Copy→Questions→Response
- Knowledge Bases: RAG: Fetch data from other data sources
- Fine-tuning: data in S3
- Unified APIs

Base Foundation Models

- Model types, performance, requirements, capabilities, compliance
- Customization level, inference options
- Inputs, outputs

Amazon Titan:

- High-Performing Foundation model from AWS
- Image, text, multi-mode, customizable with your data
- Smaller models are cost-effective

Amazon Titan: 8K Tokens
Llama: Meta 4K Tokens
Claude: 200K Tokens
Stability.AI: 77-Tokens (image only)

FM Providers: Playground

Action: Learn about them in your own time, but know the general capabilities

Compare models: select more than one model to understand capability metrics:

- Text vs Image
- Latency
- Input token
- Output token

Model Customization (Fine Tuning)

Fine-tuning: Job One Time or Continuous

- Select model→Enrich with data
- Name of the model
- Job config
- Input data:
    - Must be in AWS S3
    - Can have a validation set
- Hyperparameters:
    - How should the algorithm behave?
    - Epochs, Batch Size, Learning Rate
- Output data:
    - S3 bucket
    - Service role for access to S3
- **Must Purchase Provisioned Through Puts**

Fine-Tune A Model

- **Changes model weights (RAG DOES NOT)**
- **Must Purchase Provisioned Throughputs**
- Instruction-based: domain-specific: **labeled examples**: prompt response pairs
- Continue pre-training: **unlabeled data**: also called domain adaptation fine-tuning
- Instruction-Based Fine Tuning Sub Types:
    - Single-turn Messaging: system, message, role, content (chatbot reply)
    - Multi-turn: conversations

- Good To Know For Fine Tuning:
    - Retraining an FM requires higher budget
    - Instruction-based is likely cheaper
    - Requires an ML engineer
    - Prepare data
    - Must use provisioned throughput

Transfer Learning (Image classification/NLP)

- Re-use a pre-trained model to adapt to a **new related** task
- Pre-trained model → Transfer Learning → New Task
- General ML concept
- Fine-tuning is a specific kind of transfer learning

Fine Tuning Use Cases

- Chatbot with particular tone or persona
- More up to date information
- Training with Exclusing data

- Targeted use cases

Evaluate Model For Quality:

- Automatic Evaluation
- Built-in task types
    - Text summarization
    - Q&A
    - Text Classification
    - Open-ended text generation
- Bring your own prompt or use built-in datasets
    - Benchmark: checks bias, curated, low admin
    - Can use human evaluation: thumbs up/down, ranking, grading score
- Benchmark questions →Model to evaluate→Generated answers→Judge model
  →Benchmark answers
- Judge model looks at benchmark answer—give a grading score
    - BERT score, F1 score
- Benchmark Dataset:
    - Designed specifically for evaluating models
    - Wide range of topics
    - Helpful to measure accuracy, speed, scalability, and efficiency
    - Can help detect bias/discrimmination
    - Low admin effort to detect bias
- Human Evaluation
    - Employees, SMEs
    - Define metrics to evaluate
    - Thumbs up/down, ranks, etc.

Metrics To Evaluate an FM

ROUGE: Recall Orientated Understudy For Gisting Evaluation:

- Evaluating automatic summarization and machine translation systems
- ROUGE-N: number of matching n-grams reference vs generated
- ROUGE-L: Longest common sub-sequence between reference and gen text

BLUE: Bilingual Evaluation Understudy:

- Evaluate the quality of generated text
- Precision, penalize brevity
- Combinations of n-grams

BERTScore: Bidirectional Encoder Representations From Transformers:

- Symantec similarity between generated text (meaning)
- Pre-trained BERT models to compare contextualized embeddings of texts cosine similarity

- Captures nuances between the text
- Perplexity: how well model predicts next token—lower is better

BERT: Contextual Understanding
BLUE/ROUGE: Summarization/Translation

Business Metrics To Evaluate A Model:
- User Satisfaction
- Average Revenue Per User
- Cross-Domain Performance
- Conversion Rates
- Efficiency

RAG and Knowledge Base

Retrieval-**A**ugmented **G**eneration: A Knowledge Base
- Reference a data source outside of training data **without** being fine tuned
- Good for real-time, up-to-date information

User→Query→Search→FM→Knowedge Base→Data Source→S3

S3 →Knowledge base
User→FM Prompt → Knowledge Base → Vector Database

Vector Databases:

- Open Search Service (default)
- Aurora
- MongoDB
- Redis
- Pinecone

Embeddings Model: How To Convert The Data Into These Embdeddings Model
- Amazon Titan
- Cohere
- S3→Document Chunks→Embeddings Model→Vector Database

RAG Databases:
- Amazon OpenSearch Service (DEFAULT)
- Amazon DocumentDB
- Aurora, RDS PostGres
- Neptune

RAG Datasources:

- S3
- Confluence
- Sharepoint
- Salesforce
- Web pages

Use Cases For Amazon Bedrock:

- Customer service
- Legal Research
- Healthcare

More AI Gen Concepts

Tokenization: raw text to sequence of tokens

- Word based
- Subwords (e.g. un acceptable)

Context window: The number of tokens an LLM can consider when generating text

- larger, more info and coherence, more processing, most cost
- First factor to consider when considering a model

Embeddings: Create vectors from text, images, or audio

- Words that have similar semantics have similar embeddings
- Vectors capture many features based on one input
- Visualize with colors to show similarities
- Reduction to 2D so we can interpret

Amazon Bedrock Guardrails: Control interaction between users and foundation models

- Filter harmful or undesirable content
- Remove PII and enhance privacy
- Reduce hallucinations: answers are safe and sound
- Monitor and analyze to ensure the guardrails are good

Amazon Bedrock Agents: manage and carry out multi-step tasks:

- Perform tasks in correct order and ensure info is passed correctly
- Configured to perform specific pre-defined action groups
- Integrate with other services, APIs, DBs, etc.
- Leverage RAG when needed to retrieve information
- Instructions →API
  →Lambda functions →DB
  →Knowledge base
- Task→Agent→Model→Chain Of Thought→Steps→APIs, Search→Results→Agent

CloudWatch Integration:

Model invocation logging:

- Send logs of all invocations to S3
- Includes text, images, embeddings
- Analyze further with CW Log Insights and build alerts.
- Cloudwatch Metrics:
    - Publish metrics from Bedrock to CW
    - Can build alarms if something is “cut”

Other Features

- Bedrock studio: UI to create AI-powered apps
- Watermark detection: check if an image was generated by Amazon Titan Generator

Pricing

On-demand:

- Pay as you go
- Text Models: every input/output
- Embedding Models: every input
- Image: every image generated
- **Base models only**

Batch:

- Multiple predictions → output to single file in S3
- Discount of 50%

Provisioned Throughout:

- Purchase model units
- Guarenteed throughput
- Works with Base; **must use for Fine-tuned and Custom**

Ranked Pricing:

- $ Prompt Engineering—no model training
- $$ RAG: users external knowledge base (no retraining)
- $$$ Instruction-based Fine-tuning: requires additional computing
- $$$$ Domain Adaption Fine-tuning: model is trained on domain-specific dataset

Cost Savings:

- On-demand
- Batch 50% discount
- Provisioned throughout—not cost-saving

- Temperature: no impact on pricing
- Model size: smaller model will be cheaper
- Cost Drivers: Number of input and output tokens

AI Stylist: Application from Amazon for demo Bedrock capabilities

SECTION 6: PROMPT ENGINEERING

Naive Prompt: "Summarize what is AWS."

Prompt Engineering: develop, design, and optimize prompts:

1. Instructions: tasks
2. Context: external info to guide
3. Input Data: input for response
4. Output Indicator: type of output

Negative Prompting: explicitly instruct what ***not to* include**

- Avoid unwanted content
- Maintain Focus
- Enhanced clarify

Prompt Performance Optimization:

- System prompts: sets tone, how the model should behave
- Temperature (0 to 1): creativity of the output: cold (conservative), hot (creative, less predictable)
- Top P: The percentage of most-likely candidates that the model considers for the next token.
    - Choose a lower value to decrease the size of the pool and limit the options to more likely outputs.
    - Choose a higher value to increase the size of the pool and allow the model to consider less likely outputs.
- Top K: limit of the number of probably words. 10 = Low, 500 = High (more diverse and creative answer)
    - Choose a lower value to decrease the size of the pool and limit the options to more likely outputs.
    - Choose a higher value to increase the size of the pool and allow the model to consider less likely outputs.
- Length: max length of answer
- Stop sequences: tokens that signal the end of the prompt

Prompt Latency: how fast the model responds

- Size
- Type
- Tokens
- Not Impacted By: Top , Top K, Temperature

Other Techniques of Presenting Prompts:

- Zero-Shot Prompting: present a task ***without*** any examples or explicit training
- Few-Shots: A few examples in the prompt
- One-shot/Single-shot

- Chain of thought: Sequence of steps
- **RAG: Not a prompt engineering technique**
  Combine model capability with external data sources

Templates: simplify and standardize the process of generating prompts

- Helps with processing user input
- Orchestrate among FM, action groups and knowledge bases
- Can be used with Bedrock Agents

SECTION 7: AMAZON Q

Amazon Q Business: Gen AI Assistant for your business

- Example: Job posting, social media post, meeting topics
- Built on BedRock
- Data Connectors (fully managed RAG)
    - S3, RDS, Aurora, WorkDocs, MS 365, Salesforce, GDrive, etc.
    - Integration allows sources being crawled
- Plugins: allow Amazon Q to interact with 3rd party services
    - JIRA, ServiceNow, Zendesk, Salesforce
    - Create tickets
    - Custom plugins with API
- Access: IAM Identify Center
    - Responses generated from documents they have access to ONLY
    - Integrated with external ID providers (Active Directory, Google Login, etc.)
- Admin Controls: Used to customize responses
    - Like Guard Rails
    - Respond only with internal info
    - Global control or topic-level controls

Amazon Q Apps: Part of Q Business

- Create gen AI-powered apps without coding, using only NL
- Leverages internal data

Amazon Q Developer: Coding Assistant

- Answer questions about AWS documentation
- Answer questions about your AWS resources
- Suggest CLIs to run
- Analyze AWS bill, resolve errors and so on
- AI code companion—similar to GitHub Copilot
- Supports JavaSCript, Python, TypeScript, and C#
- Realtime code scans, bootstrapping, etc., debugging, security checks
- IDE: Visual Studio Code, Visual Studio,. Jetbrains

Amazon Q: Other Web Services

- **QuickSight**: upload dataset, ask NL questions
- **EC2**: Guidance and suggestions based on workload requirements
- **Chatbot**: Deploy chatbot in an application, e.g. slack
- **Glue**: Gen code for AWS Glue, or answer questions about scripts, errors in jobs

PartyRock: Playground: GEN AI App Builder: powered by Amazon Bedrock

SECTION 8: AI & MACHINE LEARNING

AI: Definition: the development of intelligent systems to perform a task that typically requires human intelligence:

- Perception
- Reasoning
- Learning
- Problem Solving
- Decision Making

Remember: AI⇒ML⇒Deep Learning⇒Generative AI

Data Layer→ML Framework or Algorithm→Model Layer→Application Layer (serve the model)

ML is a type of AI to build a method for machines to learn from data

- Regression
- Classification

Deep Learning: subset of ML: Deep because there are multiple layers

- Concept of neurons and synapses
- Process more complex patterns
- Input layer→hidden layers→output layer
- Computer vision, image classification, NLP
- Large input data, computationally intense
- GPU: parallel computations

Neural Networks:

- Input data→new connections→output layer
- Connections change as new data is added
- Learned

GEN AI Space:

- Unlabeled data→pretrain→foundation model→adapt→tasks
- Transformer model (LLM): process sentence as a whole
    - Transformer models work by processing input data, which can be sequences of tokens or other structured data, through a series of layers that contain ***self-attention*** mechanisms and feedforward neural networks
- Google BERT, OpenAI ChatGPT (**G**enerative Pretrained **T**ransformer)
- Diffision model: forward to noise
- Multi-modal models: multi inputs and multi outputs
  Take images, text, or audio

Terms in Exam Questions:

- **GPT**: code or text based on input prompts

- **BERT**: Bidirectional Encoder Representations from Transformers: Two directions (translations)
- **RNN**: Recurrent Nueral Network: sequential: times series, speech recognition, **VIDEOS**
- **ResNet**: Deep Convolutional Neural Network (CNN): **Images**, object detection, facial recognition
- SVM: ML algorithm for classification
- WaveNet: Audio wave forms—speech
- **GAN** (Generative Adversrial Network): ***synthetic*** data, images, sound,
- XGBoost: Gradient boosting, regression

Training Data Types:
- Labeled Data: Images of animals labeled with the type
    - Input features with output labels
    - ***Supervised learning***
- Unlabeled: input features, ***no output labels***
    - ***Unsupervised learning***
- Structured Data:
    - Tabular data
    - Time series data
- Unstructured Data
    - E.g. Multimedia content
    - Text heavy
    - Image data

Feature Engineering: domain knowledge: transform raw data into useful data

Supervised Learning:
- A mapping function to predict the output for new unseen data
- ***Needs labeled data***
- Regression: continuos
- Classification: categorical, discrete
    - Binary or multiclass/label
- KNN (nearest neighbour)
- Training (80%) vs validation (10%) vs Test set (10%)

Unsupervised: unlabeled
- Discover patterns: clustering, association rule learning, anomaly detection
- Recommendations: association rule
- Fraud detection: isolated forest/outliers

Semi-supervised: small amount of labeled data, large amount of unlabeled data
- Psuedo-labeling
- Train on labels
- Label unlabeled data

- Re-train on whole dataset

Self Supervised Learning: generate its own labels
- A lot of unlabeled data
- We want the model to generate its own labels
- Pre-text tasks: simple tasks to solve

Reinforcement Learning: Agent learns by making decisions by maximizing cumulative reward
- Gaming, Robotics, Finance, Healthcare, Autonomous vehicles
- Key Concepts:
    - Agents
    - Environment
    - Actions
    - State
    - Policy
    - Reward

Check this out: YT - @aiwarehouse

RLHF: Reinforcement Learning from Human Feedback
- Use feedback to help models learn more efficiently
- Compare model and human responses
- Base LLM→Fine Tuned LLM
- 1. Data Collection
- 2. Supervised fine-tuning of LLM
- 3. Build separate reward model
- 4. Optimize language with reward model

Model Fit
- Overfitting: good on training but poor on evaluation
- Underfitting: underperforming on training data
- Balance: some error, but following trends
- Bias: difference (error) between predicted and actual value
    - ***High bias: underfitting***: model does not match training data
    - Reduce bias: use a more complex model, more features
- Variance: how much performance changes with different data
    - **High variance: overfitting**: sensitive to changes in the training data
    - Reduce variance: Consider most important features
- HVO, HBU, LLB

Metric Evaluation:
- Confusion matrix: ***classification***
    - Actual Value vs Predicted Value
    - True positives

    - False Negative
    - False positive
    - True negative
- Confusion Metrics:
    - Precision: TP / (TP + FP) : **How many times correct about positives?**
    - Recall: TP / (TP + FN) :
    - F1 = 2 * Precision * Recall / (Precision + Recall)
    - Accuracy: TP+TN / (TP+TN+FP+FN) - **% Of Correct Predictions**
- Confusion: When to use:
    - Precision: FP are costly
    - Recall: FN are costly
    - F1 score: balance between precision and recall
    - Accuracy: best for balanced datasets
- AUC-ROC: Area under curve-receiver operator curve
    - Sensitivity vs Specificity
        - Sensitivity: True Positive Rate
        - Specificity: 1-False Positive Rate
    - Compare: True positive vs False Positive
- Regression: ***continuous value*** - regression
    - MAE: Mean Absolute Error
    - MAPE: Mean Absolute Percentage Error
    - RMSE: Root Mean Squared Error
    - R^2: Variability in model

Inferencing: makes prediction on new data

- Realtime: speed over perfection: e.g. prompt in a chatbot
- Batch: accuracy over speed: large amount of data, data analysis
- At the Edge: Less computing power like a phone
    - SLM: Local, offline, low compute
    - LLM: remote server, via APIs, must be online

Phases Of ML Project:

- Business Problem→ML Problem→Data Collection & Preparation
- Feature Engineering→Model Training & Tuning→Model Evaluation
- Did we meet the goals?
- YES: Deploy
    - Monitoring, Retrain, Iterate…
- NO: Data Collection: Feature Augmentation, Data Augmentation

Hyperparameter Tuning:

- Hyperparameter:
    - Settings to define the model and learning algorithm
    - Set before training begins
    - Learning rate, batch size, epochs, flexibility

- Hyperparameter Tuning:
    - Finding the best values
    - Improves accuracy
- How?
    - Grid search, random search
    - Sagemaker Automatic Model Tuning (AMT)
- Exam Knowledge:
    - **Learning rate**: large or small the steps are when updating weights
    - **Batch size**: how many training examples used in one interation
    - **Number of Epochs**: how many iterations
        - Too few: underfitting
        - Too many: over fitting
    - **Regularization**: balance between simple and complex
        - ***Increase: reduce*** overfitting
- Overfitting: great predictions for training dataset, but not for new data
    - Too small data
    - Too long training
    - Model complex is high, learns from noise
    - Prevention:
        - Increase training data size
        - Early model stopping
        - Data augmentation
        - Adjust hyper parameters

When is ML ***NOT*** appropriate? Deterministic problems (you may get an approximation)!

SECTION 9: AWS AI SERVICES

Comprehend: NLP

- Fully managed and serverless
- Customer interactions (email)
- Group articles by topic
- Custom classification: e.g email classification
- Named Entity Recognition (NER): extract people, places, dates, organizations, etc.
- Custom Entities: e.g. policy numbers
    - Train the model
    - Real-time or Async

Translate: Self-explanatory

- Text to text, can use a document

Transcribe: automatically convert speech into text

- Deep learning service: Automatic Speech Recognition (ASR)
- Auto-remove PII
- Multi language with custom vocabulary and custom language models
- Toxicity detection: tone (e.g. anger) and pitch, text-based cues

Poly: Opposite of Transcribe: Text→Speech

- Lexicons: AWS → Amazon Web Service
- SSML: Speech Synthsis Markup Language
- Voice Engine

Rekognition: Find objects, people, text, scenes, etc. in images and videos

- Labelling
- Context moderation: detect inappropriate content
    - Bring down the need for human review
    - Amazon A2I: for human review
    - Custom moderation adapter
- Text detection
- Face detection and analysis
- Pathing: sports
- Custom Labels: may appear in the exam
    - Label training images, upload to Amazon Rekognition, custom model

Lex: chatbots using voice or text

- Deep integration with other AWS services
- Uses LAMBDA
- Visual Builder

Amazon Personalize: real-time recommendations

- Same technology used by Amazon.com
- S3 →Amazon Personalize→Websites, Apps, SMS, Email
  Amazon Personalize API →
- Recipes: algorithms for specific use cases
    - Recommend Items
    - Personalized Rank items
    - Popular Items: Trending
    - Similar items
    - Next best action
    - Segments (item affinity)

Textract: Extract text
- Handwriting, text from scanned document
- Forms and tables
- PDFs and images
- Understands layout, forms, tables, etc.
- You can also run queries
- Use cases: expenses, ids

Amazon Kendra: Fully managed documents search service
- Sources: S3, RDS, GDrive, Sharepoint, OneDdrive, SalesForce, ServiceNow
- Knowledge index → NL search capability

Mechanical Turk: Crowdsourcing marketplace to perform a simple human task
- Eg.: 10K images you want to label, humans will tag them
- Use cases, image classification, data collection, business processing
- Integration with A2I, SageMaker Ground Truth

Amazon Augmented AI (A2I): human oversight to make sure your models are working
- Prediction high: goes to client
- Prediction low: human review, results stored in S3
    - Employees or contractors
    - Build it yourself or integrate existing service

Amazon Transcribe Medical: specific to medicine for HIPPA compliance

Amazon Comprehend Medical: specific to medicine
- From unstructured text, understand full relationships of words

Amazon HW For AI:
- GPU-based instances of EC2 (P3,P4,P5,G3,G6)
- Trainium: 100B+ parameters (Trn1): 50% cost reduction
- AWS Inferentia
    - Inference at high performance and low cost

- Trainimum and Inferentia have the lower footprint
- 4x throughout, 70% cost reduction

SECTION 10: AMAZON SAGEMAKER

End-to-end ML-managed service

- Fully managed service for developers
- All processes in one place

- Collect and prepare data
- Build and train model
- Deploy

Extra Features

- **Network isolation mode**: no outbound internet access. Cannot access S3!
- **DeepAR forecasting**: forecast time series data—leverage RNN

Built-in Algorithms:
- Supervised
- Unsupervised
    - PCA
    - K-means
    - Anomaly detection
    - Textual Algorithms
    - Image processing

Automatic Model Tuning: AMT: Optimize for the objective metric

Model deployment: one-click
- Auto scaling
- Reduced overhead
- **Real-time:** end point
- **Serverless**: select RAM we need
    - Tolerate more latency
- **Asynchronous**: large payloads: near real-time latency requirements
- **Batch**: prediction for entire dataset

Inference Type	Latency	Payload Size	Processing Time	Use Case
Real-time Inference	Low (milliseconds to seconds)	Up to 6 MB (one record)	Max 60 seconds	Fast, near-instant predictions for web/mobile apps
Serverless Inference	Low (milliseconds to seconds)	Up to 4 MB (one record)	Max 60 seconds	Sporadic, short-term inference without infrastructure, can tolerate cold starts
Asynchronous Inference	Medium to High "near real-time"	Up to 1 GB (one record)	Max 1 hour	Large payloads and workloads requiring longer processing times
Batch Transform	High (minutes to hours)	Up to 100 MB per invocation (per mini batch)	Max 1 hour	Bulk processing for large datasets Concurrent processing

SageMaker Studio: interface for end-to-end ML developments

- Team collaboration
- Deploy ML
- Automate workflows

SageMaker Data Wrangler: Prepare data for SageMaker

- Preparation, transformation, feature engineering
- Selection, exploration, visualization
- SQL support and quality tool

SageMaker Feature Store:

- Ingest features from a variety of sources
- Publish directly from Data Wrangler
- Discoverable from SageMaker studio

SageMaker Clarify: Compare models

- Evaluate Foundational Models (Involve humans)
- Built-in models and algorithms
- Model Explainability: a set of tools to explain why models make predictions
- Increase trust and understanding of model
- Detect Bias: data sets and model

SageMaker GroundTruth: RLHF: Reinforcement Learning From Human Feedback

- Model review and customization
- Human feedback for ML
- GroundTruth Plus: Use workforce to carry out data labelling

SageMaker LM Governance:

- Model Cards: essential information on the model
- Model Dashboard: view all your models, insights and information

- Role Manager:
- Model Monitor: setup once model is in production
    - Continuous vs Schedule
    - Deviation in quality can send an alert
- Model Registry: track, manage and **version** ML models
- Pipelines: create a workflow to automate the process of building, training and deploying
    - CI/CD
    - Steps: Processing, Training, Tuning, AutoML, Model, ClarifyCheck, QualityCheck

SageMaker Jumpstart:

- ML hub to find pre-trained foundation models
    - Browse→Experiment→Customize→Deploy
- ML solutions
    - Access→Select, Customize→Deploy

SageMaker Canvas: Build ML models using visual interface

- Has ready-to-use models from Bedrock or Jumpstart
- AutoML, SageMaker Autopilot

MLFlow: open-source tool to manage entire ML lifecycle

- lets you create, manage, analyze, and compare your machine learning experiments
- MLFlow Tracking Server

SECTION 11: AI RESPONSIBILITY & GOVERNANCE

Responsible AI

- Trust, transparency, mitigating risk
- Security: confidentiality, integrity
- Governance: add value and manage risk with clear policies and guidelines, align legal and regulatory requirements
- Compliance: adherence to regulation

Core Dimensions:
- Fairness
- Explainability
- Privacy/Security
- Transparency
- Veracity, Robustness
- Governance
- Safety
- Controllability

AWS Services To Help

- Bedrock:
    - Human and auto-model evaluation
    - Guardrails
- SageMaker Clarify
    - FM evaluation
    - Bias detection
- Sage Maker Data Wrangler: Augment data
- Sage Model Monitor: quality analysis
- Amazon Augmented AI (A2I): Human review of predictions
- Governance:
    - SageMaker Role Manager: security at user level
    - SageMaker Model Cards
    - SageMaker Model Dashboard

AWS AI Service Cards:
- Responsible AI Documentation
- For Of Responsible AI with use cases, deployment best practices

Interpretability: A human can understand the cause of a decision in a ML model
- Access to a system
- Answer why and how

- High transparency⇒High Interpretability⇒Poor performance
- Explainability: understand the nature and behaviour of the model
    - Look at inputs; **explain outputs**
    - E.g., decision trees
- PDP: Partial Dependency Plots: Black box models: Change one parameter only
    - the dependence of the predicted target response on a set of input features of interest
- Shapley values: marginal effect: determine the contribution that each feature made to model predictions
- HCD: Human Centered Design:
    - Design for amplified decision-making: clarity, simplicity, usability
    - Design for unbiased decision-making:
    - Design for human and AI learning

AI Challenges

- Regulatory violations
- Social risk
- Data security and privacy
- Toxicity
    - Toxic content vs censorship
    - Mitigation: curate training data and use guardrails
- Hallucinations: assertions or claims that sound true
    - Mitigation: educate users, ensure verification
- Plagiarism: concern about writing essays, job applications, etc
    - Difficulties with LLM tracing
- Non Deterministic
- Prompt Misuses: intentional introduction of malicious content or biased data
    - Prompt injection: influencing the outputs by embedding specific instructions
    - Hijack model behaviour
    - Exposure: confidential info that can be revealed
    - Prompt leaking:
    - Jailbreaking: trained with constraints
        - Many shot vs Few shot

Compliance

Some industries require additional rules for compliance

- Financial services
- Health care
- Aerospace

Challenges:

- Complex and opaque

- Dynamism and adaptability
- Emergent capabilities
- Unique risks: bias
- Algorithm accountability: fairness

AWS has a lot of compliance certifications—140+

Model Cards: standardized documentation format

Governance Framework

- Establish a governance board
- Define roles and responsibilities
- Implement policies and procedures

AWS Tools:

- AWS Config
- Amazon Inspector
- AWS Audit Manager
- AWS Artificate
- AWS CloudTrail
- AWS Trusted Advisor

Strategies:

- Policies
- Review Cadence
- Review Stategies
- Transparency Standards
- Team Training

Data Governance Strategies:

- Responsible AI
- Structures & Roles
- Data sharing and collaboration

Data Management Concepts

- Lifecycles
- Logging
- Residency
- Monitoring
- Analysis
- Retention: regulatory requirements
- Lineage: source, origins, cataloging

Security & Privacy

- Threat detection
- Vulnerability management
- Infrastructure protection
- Prompt injection
- Data Encryption

Monitoring

- Performance
    - Accuracy
    - Precision
    - Recall
    - F1-score
    - Latency
- Infra-structure
    - Compute
    - Network
    - Storage
- Bias and fairness

Shared Responsibility Model

Secure Data Engineering

- Assessing data quality
- Privacy-enhancing technologies
- Data access control
- Data Integrity

Gen AI Security Scoping Matrix: Five Scopes:

Scope 1: Consumer App: E.g., ChatGPT
Scope 2: Enterprise App: E.g., Salesforce, Amazon Q Developer
Ssope 3: Pre-trained Model: E.g., Amazon Bedrock
Scope 4: Fine-tuned Model: E.g., SageMaker
Scope 5: Self-trained Model: E.g., SageMaker

MLOps (Dev Ops)

Practice to deploy

Key Principles:

- Version control
- Automation

- CI/CD
- Continuos retraining
- Continuos monitoring
- Data preparation⇒Model Build⇒Model Eval⇒Model Selection⇒Deployment⇒Monitoring

SECTION 12: NOTES FROM MOCK EXAMS:

First Exam: 69%
Re-take: 78% (used notes)
Re-take: 86%
Re-take: 81%
Re-take: 73%

Generative AI:

Amazon Bedrock: **choose** the underlying Foundation Model
Amazon Q: **CANNOT** choose the underlying Foundation Model

Hyperparameters:

Epochs: increase accuracy by increasing the number of epochs, allowing the model to learn from the training data for a longer period
Learning Rate: for text models, a small increases mean more accurate models, takes longer
Batch size: number of records for each interval sent to each GPU
Regularization: Increasing regularization is beneficial when the model is **overfitting**

Model parameters: values that define a model and its behavior in interpreting input and generating responses.

Hyperparameters: values that can be adjusted for model customization to control the training process

Agents For Amazon Bedrock:

Agents for Amazon Bedrock are fully managed capabilities that make it easier for developers to create generative AI-based applications that can complete complex tasks for a wide range of use cases and deliver up-to-date answers based on proprietary knowledge sources.

Prevent Overfitting:

Early Stopping: stop before the model learns the noise. Timing is tricky.
Pruning: prioritize the features or parameters that most impact the final prediction
Regularization: grading features based on importance in prediction
Ensembling: combining predictions from several ML algorithms: bagging and boosting
Data Augmentation: changes sample data slightly to make training data unique

Shapely and PDP:

Shapley values:	**local** explanation: quantify contribution of each feature to prediction individual predictions
PDP:	**global** explanation: marginal effect of a feature on the model's predictions model's behavior at a dataset level.

Model Monitoring:

Amazon SageMaker Dashboard & Model: Monitor
Amazon SageMaker Clarify: BIAS

Model Improvement:

Transfer Learning: allows a model to utilize the knowledge learned from one task or dataset to improve its performance on a new, but related task. **Multiple model improvement**.

Incremental Training: Enhance a single model's performance with its own data.

Reinforcement Learning: not for model optimization and improvement.

Self-Supervised Learning: Foundation training, not model optimization and improvement.

Supervised Learning:

- Logistic Regression
- Linear Regression
- Decision Tree
- Neural Network

Semi-Supervised: DFS

- Document Classification
- Fraud Identification
- Sentiment Analysis

Unsupervised: AC/DP

- Association rule learning
- Clustering
- Dimensionality reduction
- Probability Density

RAG: Databases: Open Search Vector Database:

Opensearch is specifically built to handle search and analytics workloads, including fast index lookups and similarity scoring. OpenSearch supports full-text search, vector search, and advanced data indexing, which are essential for the Retrieval-Augmented Generation (RAG) framework

Context Window: Amount of text a model can consider
Tokens: Items of text, what the context window is measuring

Foundation Models:

Foundation models use ***self-supervised*** learning to create labels from input data. This means no one has instructed or trained the model with labeled training data sets.

Fine-tuning an FM is a **supervised learning** process

Image Processing: Involves straightforward algorithms and operations that can be executed without learning from data. Tasks like blurring, sharpening, and color adjustments are examples of image processing tasks.

Computer Vision: Employs complex algorithms that require learning from data sets

Token: individual units of text (words, subwords, or characters)
Embeddings: numerical representations of tokens (Vectors)

Explainability: focuses on explaining the results of a model
Interpretability: focuses on understanding how the model works.

Amazon Augmented AI (Amazon A2I):

Amazon Augmented AI (A2I) is a service that helps implement human review workflows for machine learning ***predictions***. It integrates human judgment into ML workflows, allowing for ***reviews and corrections*** of model predictions, which is critical for applications requiring ***high accuracy and accountability***.

AWS IAM service: RESOURCES
IAM Identity Center: Users

Multi-modal *EMBEDDING* model: represent and align different types of data, understand and interpret both forms of input simultaneously

Multi-modal *GENERATIVE* model: complex and typically used for generating new content

Global Services:

- AWS Identity and Access Management (AWS IAM)
- Amazon CloudFront
- Amazon Route 53
- AWS Web Application Firewall (AWS WAF)

BENCHMARK: Check for BIAS in LLMs

Model Customization: Model Weights: Fine-tuning, model customization
Model Response: RAG: Customize a response with up to date data

Hijacking: manipulate to serve malicious purposes

Jailbreaking: bypass built-in restrictions

Trainium: DL training
Inferentia: DL applications (inference)

Textract: Image scans of documents

BERT: Bidirectional Encoder Representations from Transformers: specifically designed for understanding the **context** of words in a sentence. BERT uses DL to predict missing words

DEEPRACER: Wifi-enabled 1/18th scale race car on simulator—Reinforcement Learning

Amazon SageMaker Automatic Model Tuning: Does it all, nothing mandatory

Amazon SageMaker Ground Truth: Building training datasets
Amazon Augmented AI: Human review of predictions

Multi-label classification: Train models and classify documents with more than one label
Multi-class classification: Assigning a data point to one category out of several possible

Amazon Comprehend: Detect and redact PII in customer emails, support tickets, etc.

Benchmark: standardized sets of labeled image for model comparison & evaluation

RAG: can enhance response accuracy
Prompt Eng: can tailor responses