AWS ML Associate — Section 1.3: Train Models

This section covers how to train machine learning models in AWS SageMaker, including choosing algorithms, configuring jobs, and understanding modes of training (built-in vs custom, single vs distributed). It focuses on the end-to-end model training process.

# Training Options in SageMaker

• Built-in Algorithms — optimized for speed and scale (e.g., XGBoost, Linear Learner, Image Classification)

• Pre-built Containers — popular frameworks (TensorFlow, PyTorch, SKLearn)

• Custom Containers — bring your own Docker image with custom code/libraries

• SageMaker Autopilot — automated model selection, training, and tuning

# Training Concepts (Exam Focus)

• Input data comes from S3 channels (Pipe vs File mode)

• Choose correct algorithm based on data type (structured, text, image, etc.)

• Training jobs run in managed compute environments (instances, clusters)

• Hyperparameters affect model performance (tuning handled in 1.4)

• Distributed training for large datasets (parameter servers, Horovod for deep learning)

# Experiment Tracking

• SageMaker Experiments — organize and track multiple training runs

• MLflow integration possible for logging metrics and artifacts

• Metrics: accuracy, precision, recall, F1, AUC depending on task

# Exam Tips

• Always load data from S3 — SageMaker training jobs cannot train directly from RDS/Redshift

• Built-in algorithms scale better than custom scripts; use them if they meet requirements

• Autopilot = automated training + model selection; great for quick baseline models

• For distributed deep learning, remember Horovod (TensorFlow/PyTorch) vs parameter servers

• Understand which metric to optimize for classification vs regression