AWS ML Associate — Sections 1.1 to 1.3 Summary

# Section 1.1: Collect, Ingest, and Store Data

This section covers how to bring data into AWS, store it securely, and prepare it for ML workflows in SageMaker.

## Data Sources

• Batch files: CSV, Parquet, JSON, logs

• Databases: on‑prem, Amazon RDS/Aurora (SQL), DynamoDB (NoSQL)

• Streaming: IoT devices, clickstreams, apps

• External: SFTP partners, SaaS exports

## Ingestion Tools

• Kinesis Data Streams / Firehose — real-time streaming to S3/Redshift

• AWS Glue — ETL, schema discovery, job orchestration

• Database Migration Service (DMS) — move relational DBs into AWS

• AWS Transfer Family — managed SFTP/FTP for file ingestion

• Snowball — bulk offline ingestion

## Storage (Landing & Curated)

• Amazon S3 — primary ML data lake, versioning, lifecycle mgmt

• Amazon Redshift — data warehouse, Spectrum queries on S3

• Amazon RDS/Aurora — relational queries

• Amazon DynamoDB — NoSQL key-value for features

## Catalog & Query

• Glue Data Catalog — schemas, partitions, metadata

• Athena / Redshift Spectrum — SQL queries on S3

• Lake Formation — fine-grained permissions

## Security & Governance

• IAM roles, bucket policies — access control

• Encryption — SSE-S3 (managed), SSE-KMS (customer keys)

• Network isolation — VPC endpoints, PrivateLink

## SageMaker Consumption

• Training input channels from S3 — Pipe (stream) vs File (copy)

• Processing jobs / Data Wrangler — feature engineering & prep

• Ground Truth — managed data labeling service

• Model Registry — manage versions, deploy for batch/real-time inference

## Exam Tips

• Default landing zone is S3; always think S3 → Glue → Athena/Redshift → SageMaker

• Kinesis/Firehose for streams; DMS for DB migrations; Glue for ETL

• Prefer Pipe mode for large datasets in SageMaker

• Always consider cost (S3 cheapest), scalability, and encryption by default

# Section 1.2: Transform and Prepare Data

This section focuses on cleaning, transforming, and preparing datasets so they’re ready for ML workflows in SageMaker.

## Data Transformation & Preparation Tools

• AWS Glue DataBrew — no-code visual data prep tool for profiling, transformations, and cleaning

• AWS Glue ETL Jobs — Python/Scala ETL scripts for large-scale transformations

• SageMaker Processing — run preprocessing code (Pandas, Spark, SKLearn) inside managed containers

• SageMaker Data Wrangler — visual tool for exploration and feature engineering, integrates with pipelines

## Data Preparation Concepts (Exam Focus)

• Data Cleaning — handle missing values (drop, impute), remove duplicates, normalize/standardize values

• Feature Engineering — encoding (one-hot, embeddings), scaling (min-max, z-score), text feature extraction (TF-IDF, embeddings)

• Splitting Data — train/validation/test (70/15/15 or 80/20), stratified sampling for classification

• Data Quality — detect imbalance or skew, apply oversampling/undersampling/SMOTE

## Integration with SageMaker

• Processing Jobs — scalable preprocessing before training

• Pipelines — automate transform → train → evaluate → deploy

• Feature Store — central repository for consistent features across training/inference

## Exam Tips

• Data Wrangler = preferred tool for feature prep in SageMaker

• Glue DataBrew = no-code prep; Glue ETL = large-scale code-based prep

• SageMaker Processing Jobs = flexible, scalable preprocessing environment

• Always split data properly to prevent leakage (train/val/test)

• Normalize/standardize numerical values for scale-sensitive algorithms

• Stratified sampling helps with class imbalance

• Use Feature Store for consistency across training and serving

# 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).

## 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