Here is the full **syllabus outline** for the AWS Certified Machine Learning Engineer – Associate (MLA-C01) study guide, based on the official exam guide. The content is structured by domain → task → topic, and includes all “Skills in:” points as subtopics.

📚 AWS Certified Machine Learning Engineer – Associate (MLA-C01) – Study Guide Syllabus

---

🔵 Domain 1: Data Preparation for Machine Learning (28%)

Task 1.1: Ingest and Store Data

Topics:

1. Extracting data from AWS storage:
   - Amazon S3
   - Amazon EBS
   - Amazon EFS
   - Amazon RDS
   - Amazon DynamoDB
   - Using S3 Transfer Acceleration and EBS Provisioned IOPS
2. Choosing appropriate data formats:
   - Parquet
   - JSON

   - CSV
   - ORC

3. Ingesting data into SageMaker:
   - SageMaker Data Wrangler
   - SageMaker Feature Store
4. Merging data from multiple sources:
   - Programming (Python/Pandas)
   - AWS Glue
   - Apache Spark
5. Troubleshooting ingestion and storage issues:
   - Capacity bottlenecks
   - Scalability limitations
   - Debugging strategies
6. Making initial storage decisions:
   - Cost
   - Performance
   - Data structure trade-offs

Task 1.2: Transform Data and Perform Feature Engineering

Topics:

1. Data transformation with AWS tools:
   - AWS Glue

   - AWS Glue DataBrew
   - Amazon EMR (Spark)
   - SageMaker Data Wrangler
2. Feature creation and management:
   - Using SageMaker Feature Store
   - Feature naming/versioning
3. Data labeling and validation:
   - Amazon SageMaker Ground Truth
   - Amazon Mechanical Turk

Task 1.3: Ensure Data Integrity and Prepare for Modeling

Topics:

1. Data validation:
   - AWS Glue DataBrew
   - AWS Glue Data Quality
2. Identifying and mitigating bias:
   - Types of bias (selection, measurement)
   - Using SageMaker Clarify
3. Reducing prediction bias:
   - Splitting
   - Shuffling
   - Data augmentation

4. Configuring training data storage:
    - Amazon EFS
    - Amazon FSx

🟡 Domain 2: ML Model Development (26%)

Task 2.1: Choose a Modeling Approach

Topics:

1. Assessing data/problem feasibility
2. Selecting ML models or algorithms
3. Choosing built-in algorithms, foundation models, and templates:
    - SageMaker JumpStart
    - Amazon Bedrock
4. Selecting models based on cost
5. Choosing AWS AI services:
    - Amazon Rekognition
    - Amazon Translate
    - Amazon Transcribe
    - Amazon Bedrock

Task 2.2: Train and Refine Models

Topics:

1. Using SageMaker built-in algorithms
2. Using script mode for TensorFlow, PyTorch
3. Fine-tuning pre-trained models
4. Hyperparameter tuning:
    - SageMaker AMT
    - Bayesian optimization
5. Automated hyperparameter optimization
6. Preventing overfitting/underfitting:
    - Regularization
    - Feature selection
7. Combining models:
    - Ensembling
    - Boosting
    - Stacking
8. Reducing model size:
    - Compression
    - Pruning
    - Feature removal
9. Model versioning:
    - SageMaker Model Registry

Task 2.3: Analyze Model Performance

Topics:

1. Evaluation metrics:
    - Classification: Accuracy, F1, Precision, Recall, ROC-AUC
    - Regression: RMSE, MAE
2. Tradeoffs between cost, performance, time
3. Running reproducible experiments
4. Shadow testing: Production vs variant
5. Using SageMaker Clarify
6. Debugging convergence issues:
    - SageMaker Model Debugger

🟣 Domain 3: Deployment and Orchestration of ML Workflows (22%)

Task 3.1: Select Deployment Infrastructure

Topics:

1. Performance/cost/latency evaluation
2. Compute environments:
    - CPU, GPU, ARM, bandwidth, memory
3. Deployment orchestrators:
    - Apache Airflow
    - SageMaker Pipelines

4. Multi-model/multi-container deployments
5. Deployment targets:
   - SageMaker Endpoints
   - Amazon ECS
   - Amazon EKS
   - AWS Lambda
6. Model deployment strategies:
   - Real-time
   - Batch
   - Async
   - Serverless

Task 3.2: Script and Manage Infrastructure

Topics:

1. Best practices for scalable, cost-effective ML infra
2. Infrastructure automation:
   - AWS CloudFormation
   - AWS CDK
3. Container management:
   - Amazon ECR
   - Amazon EKS
   - Amazon ECS

    - BYOC
4. SageMaker VPC configuration
5. SageMaker SDK for deployment
6. Auto scaling metric configuration:
    - CPU, latency, invocation

Task 3.3: Set Up CI/CD Pipelines

Topics:

1. Configuring CodeBuild, CodeDeploy, CodePipeline
2. Using GitHub Flow / Gitflow structures
3. Automating model build/deploy:
    - SagePipelines
    - EventBridge
4. Creating automated tests:
    - Integration
    - Unit
    - E2E
5. Retaining model retraining mechanisms

🟢 Domain 4: ML Solution Monitoring, Maintenance, and Security (24%)

Task 4.1: Monitor Model Inference

Topics:

1. Monitoring production models:
   - SageMaker Model Monitor
2. Workflow anomaly detection
3. Detecting distribution shift:
   - SageMaker Clarify
4. A/B testing for model performance

Task 4.2: Monitor and Optimize Infrastructure and Costs

Topics:

1. Troubleshooting and monitoring tools:
   - CloudWatch
   - CloudTrail
   - X-Ray
   - QuickSight
2. Creating CloudTrail trails and EventBridge alerts
3. Rightsizing instances:
   - SageMaker Inference Recommender
   - AWS Compute Optimizer
4. Capacity management:
   - Auto scaling

    - Provisioned concurrency

5. Cost optimization:
    - Spot vs On-Demand vs Reserved
    - Budgets
    - Cost Explorer
    - Trusted Advisor

Task 4.3: Secure AWS Resources

Topics:

1. IAM configuration:
    - Roles, policies, groups
    - Least privilege access
2. Security tools:
    - SageMaker Role Manager
3. Auditing/logging ML systems
4. VPCs and subnet security