



MLA-C01 Domain 1: Data Preparation for ML

AWS MLA-C01 exam guide tasks 1.1-1.3, SageMaker AI developer guide, as of 2026 | Ingest and store training data, formats and input modes, transform and engineer features (Data Wrangler in Canvas, Processing, Feature Store), labelling, data integrity, bias checks with Clarify, splits

MLA-C01 Domain 1 28 % SageMaker AI Feature Store Clarify

Task 1.1: ingest and store training data

Domain 1 is the heaviest MLA-C01 domain at **28 %**. Data engineering basics (Kinesis, Glue, Athena, Redshift) are taught in the DEA-C01 chapter; this sheet keeps what an ML engineer adds: formats and input modes for training, feature preparation, labelling, and checks for quality, leakage and bias.

Format	Good for	SageMaker AI notes
CSV	tabular built-in algorithms	XGBoost and Linear Learner expect the target in the first column and no header
Parquet	large tabular data, Spark, Athena	columnar, compressed; XGBoost and Autopilot accept it
RecordIO-protobuf	built-in algorithms in Pipe mode	native for Linear Learner, k-means, PCA, Factorization Machines
JSON Lines	text, inference payloads, Ground Truth manifests	one JSON object per line
Images (JPEG/PNG) + manifest	vision	augmented manifest file lists S3 URLs and labels
TFRecord / Web-Dataset	TensorFlow / PyTorch at scale	shards streamed by the framework

File mode (default) — Copies the whole channel from S3 to the instance's EBS volume before training starts; simple, slow for TBs.

FastFile mode — Exposes S3 objects as files that stream on first read: no download wait, works with existing file-reading code.

Pipe mode — Streams data through a Unix pipe (RecordIO, CSV, Parquet support depends on the algorithm); legacy but still tested.

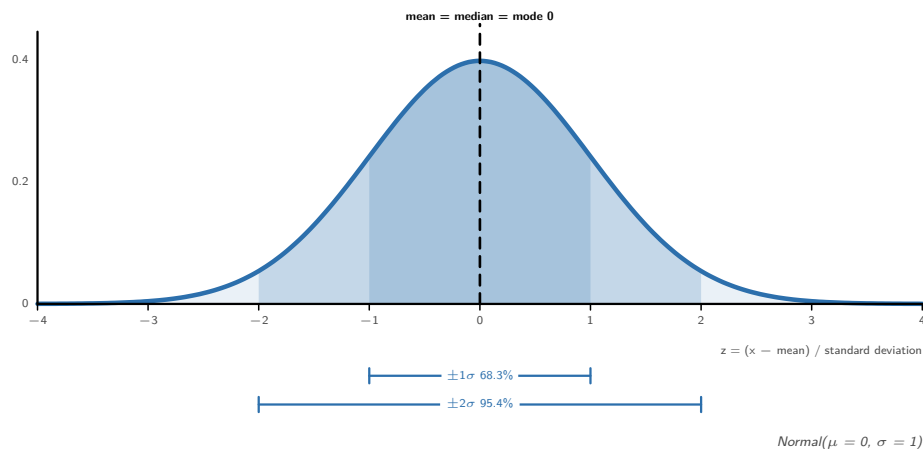
FSx for Lustre / EFS input — File system channel for repeated epochs over very large datasets with high throughput, linked to an S3 bucket (Lustre).

EXAM TIP

'Training takes too long to start because the dataset is copied from S3' → FastFile (or Pipe) mode. 'Many training jobs read the same 10 TB repeatedly' → FSx for Lustre linked to S3. 'Stream clickstream features for real-time inference' → Kinesis + Feature Store online store.

Task 1.2: transform data and engineer features

Problem in the data	Technique	Watch out
Missing values	drop rows, impute mean/median/mode, add a 'was missing' flag, model-based imputation	impute with statistics from the training split only
Outliers	cap (winsorise), log transform, robust scaler, remove if errors	fraud and anomalies ARE the outliers
Skewed numeric feature	log or Box-Cox transform, binning	check the test distribution too
Different scales	standardise (z-score) or min-max normalise	needed for k-NN, k-means, linear models, neural nets; not for trees
Categories, low cardinality	one-hot encoding	dimension explosion above a few dozen values
Categories, high cardinality	target/frequency encoding, hashing, embeddings	target encoding leaks without out-of-fold fitting
Ordered categories	ordinal (label) encoding	only when order is meaningful ($S < M < L$)
Text	tokenise, TF-IDF, embeddings (BlazingText, Bedrock Titan)	fit vocabulary on train only
Dates	extract day of week, hour, holidays, cyclical sin/cos	time zones



Why standardise: after z-scoring, about 95 % of a normal feature lies within ± 2 , so features share one scale

Tool	Use it when
SageMaker Canvas Data Wrangler	visual exploration and 300+ transforms; export the flow to a Processing job or a Pipelines step
SageMaker Processing (SKLearnProcessor, PySparkProcessor)	scripted, repeatable preprocessing inside an ML pipeline
AWS Glue / Glue DataBrew	large ETL shared with the data team; DataBrew for no-code cleaning
Amazon EMR / EMR Serverless	Spark at very large scale
SageMaker Feature Store	share features between teams, consistent online and offline values, point-in-time training sets

```
Feature Store: create a group, ingest, read online
from sagemaker.feature_store.feature_group import FeatureGroup
fg = FeatureGroup(name="customers", sagemaker_session=sess)
fg.load_feature_definitions(data_frame=df) # infer types
fg.create(s3_uri="s3://ml-features/offline",
          record_identifier_name="customer_id",
          event_time_feature_name="event_time",
          role_arn=role, enable_online_store=True)
fg.ingest(data_frame=df, max_workers=4, wait=True)
rt = boto3.client("sagemaker-featurestore-runtime")
rt.get_record(FeatureGroupName="customers",
              RecordIdentifierValueAsString="c-42")
```

Task 1.3: data integrity, labelling and bias

Column	Type	Missing	Distinct	Distribution	Issues
customer_id	id	0%	240.0k		drop before training
tenure_months	int	1.0%	120		
monthly_charges	float	0%	15.0k		
plan	categorical		4	basic, plus, pro, legacy	
cancel_reason	categorical	86%	9		leakage: filled after churn, mostly missing
churned	bool		2	0, 1	10 % positives: imbalance

Training table audit before the first job: each flag is a task for the preparation step

Class imbalance (Clarify pre-training bias metric CI)

FORMULA $CI = (na - nd) / (na + nd)$

WHERE

- CI = class imbalance between two facet groups; 0 = balanced, ± 1 = only one group present (-1 to +1)
- na = number of examples in the advantaged facet value (for example age < 60) (rows)
- nd = number of examples in the disadvantaged facet value (age ≥ 60) (rows)

WORKED

```
na = 200000
nd = 40000
CI = (200 000 - 40 000) / (200 000 + 40 000)
CI = 0.6667
```

RESULT CI = 0.6667

READ IT 0.67: the dataset holds far more younger customers, so the model may learn less about older ones; rebalance or collect more data before training

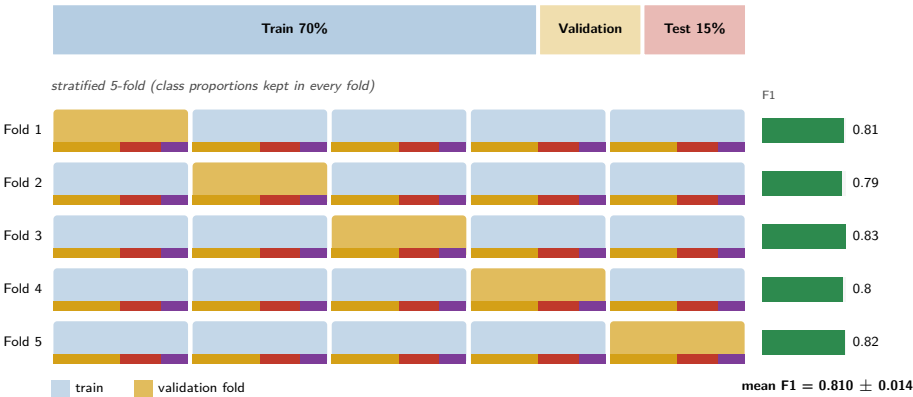
MISTAKES

- CI looks at the facet (a sensitive attribute), not at the label imbalance
- Compute it on the training split that the model really sees
- Other pre-training metrics: DPL (difference in proportions of labels), KL, JS divergence

Integrity issue	Fix	AWS tool
Label imbalance (10 % positives)	class weights (scale_pos_weight in XGBoost), oversampling, SMOTE, undersampling, more data	Data Wrangler balance transform, Processing
Leakage	remove post-event columns, split by time, fit transforms on train only	Data Wrangler target-leakage insight
Duplicates across splits	deduplicate before splitting, group split by entity	Processing, Glue
Unlabelled data	human labelling with active learning	Ground Truth, Ground Truth Plus, Augmented AI for review

PII in training data	detect, drop or mask before training	Macie (S3), Comprehend PII detection, Glue sensitive data
Data quality drift between runs	rules and baselines	Glue Data Quality, DataBrew, Model Monitor baseline

Dataset



Splits for the churn model: hold out a test set once, tune with stratified 5-fold to keep 10 % positives in every fold

✓ Myths, recap and self-test

YOU MAY HEAR
"Tree models need standardised features"

BUSTED

✓ **ACTUALLY**
Trees split on thresholds, scale does not matter
Scale for k-NN, k-means, linear models and neural networks.

YOU MAY HEAR
"A random split is always fine"

BUSTED

✓ **ACTUALLY**
Time series and repeated entities need time or group splits
A random split lets the future or the same customer leak into validation.

YOU MAY HEAR
"Oversampling before the split is harmless"

BUSTED

✓ **ACTUALLY**
It copies rows into validation and inflates scores
Split first, then resample the training set only.

YOU MAY HEAR
"Clarify only works after training"

BUSTED

✓ **ACTUALLY**
Clarify also measures pre-training bias on the dataset
CI and DPL run on data before any model exists.

Stream, do not copy
FastFile/Pipe, FSx for Lustre

Fit transforms on train only
then apply to validation and test

Feature Store
same features online and offline

Hunt leakage and imbalance
drop post-event columns, weight classes

Bias before training
Clarify CI, DPL

Q1 Where must the label be for SageMaker XGBoost CSV training data?

A First column, no header row.

Q2 Low-latency feature lookups at inference and point-in-time training sets from the same definitions?

A SageMaker Feature Store with online and offline stores.

Q3 Visual data preparation with leakage and imbalance insights?

A Data Wrangler in SageMaker Canvas.

Q4 Reduce labelling cost for 500,000 images?

A Ground Truth with automated data labelling (active learning) and a human workforce for low-confidence items.

Q5 Which encoding for a 'country' column with 5 values?

A One-hot encoding.

Q6 Which XGBoost parameter handles class imbalance?

A scale_pos_weight (about negatives / positives).

Q7 Training start is slow because 2 TB is downloaded first: fix?

A FastFile input mode (or Pipe mode).

Q8 Name two Clarify pre-training bias metrics.

A Class imbalance (CI) and difference in proportions of labels (DPL).

Q9 Why split time-series data by date?

A A random split leaks future information into training; validate on later periods.