Implementing a Machine Learning Pipeline with Amazon SageMaker

PROBLEM FRAMING · DATA COLLECTION & SECURITY · EVALUATION · FEATURE ENGINEERING · TRAINING · HOSTING · METRICS · TUNING

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

01 **The pipeline is iterative:** the stages run **problem formulation → collect & label data → evaluate data → feature engineering → select & train → deploy → evaluate → tune**, then loop back with feature or data augmentation until the model *meets the business goal*. Real problems take many passes.

02 **Frame before you build:** convert the business request into an ML problem — ask **why** until the goal is solid, move from qualitative to *measurable* (quantitative) success, then decide the ML type. Most problems are **classification** (does it belong to a class?) or **regression** (predict a numeric value).

03 **Observations = target + features:** supervised learning needs many **labeled** observations where the answer is known. The **target** is what you predict; a **feature** is an attribute used to find patterns. Data must be **representative** — collect both positive and negative examples or the model can't learn the rare class.

04 **Split to avoid overfitting:** never evaluate on training data. Use a **holdout** split (train / validation / test, e.g. 80/10/10 or 70/15/15) or **k-fold cross validation** for small datasets. **Shuffle** (or *stratify*) first so order doesn't bias the model. For SageMaker CSV: **no header row, target in the first column**.

05 **Match the metric to the goal:** the **confusion matrix** (TP/FP/FN/TN) yields **sensitivity** (recall), **specificity**, **precision**, **F1**, and **accuracy**; use **AUC-ROC** to compare models. Accuracy misleads when true negatives dominate. For regression, use **mean squared error** and **R^2**.

02 KEY TERMS

TERM	DEFINITION
Observation	One row of labeled data used to train a supervised model; made up of a target plus features
Target	The answer you want the model to predict (for example, fraud or not fraud)
Feature	An attribute of an observation used to identify patterns that predict the target
ETL	Extract, transform, load — pull data from many sources, modify/combine it, and load it into one repository (such as Amazon S3)
AWS Glue	Fully managed, serverless ETL service; crawlers build a Data Catalog and an engine generates Python or Scala code
Feature engineering	Feature selection (keep the most relevant features) plus feature extraction (build new features from raw data)
Encoding	Converting categorical (non-numeric) data to numbers; ordinal keeps order (1,2,3), non-ordinal splits into multiple columns
Overfitting	When a model memorizes the particulars of its training data instead of learning general relationships between features and labels
Holdout method	Splitting data into separate training, validation, and test sets to measure real-world performance
K-fold cross validation	Partition data into K segments, rotate which segment is validation, average results — uses all data for training
Hyperparameter	A knob that tunes the algorithm (model, optimizer, or data) and is set before training, not learned from data

TERM	DEFINITION
Confusion matrix	A table comparing predicted vs. actual classes as true/false positives and negatives (TP, FP, FN, TN)

03 THE MACHINE LEARNING PIPELINE

iterative — loop until it meets the business goal

01 Problem formulation → **02** Collect & label data → **03** Evaluate data → **04** Feature engineering → **05** Select & train model → **06** Deploy model → **07** Evaluate model → **08** Tune model

04 CLASSIFICATION METRICS FROM THE CONFUSION MATRIX

the metric must match the business goal

METRIC	WHAT IT MEASURES	FAVOR IT WHEN
Sensitivity	TP ÷ actual positives — share of positives caught (recall, hit rate, TPR)	Missing a positive is costly (disease, fraud)
Specificity	TN ÷ actual negatives — share of negatives caught (TNR)	Wrongly flagging a negative is costly
Precision	TP ÷ (TP + FP) — positive predictions that are actually correct	A false positive is expensive (spam filter)
Accuracy	(TP + TN) ÷ all predictions — the model's overall score	Classes are balanced (misleads with many TNs)
F1 score	Combines precision and sensitivity into one number	Class imbalance, but keep the two in balance
AUC-ROC	Area under the ROC curve (sensitivity vs. false-positive rate)	Quickly comparing whole models to each other

05 AMAZON SAGEMAKER BUILT-IN ALGORITHMS

know which algorithm fits the problem type

CATEGORY	ALGORITHM(S)	TYPICAL USE
Classification	XGBoost, Linear learner	Fraud yes/no; binary or multiclass labels
Regression (quantitative)	XGBoost, Linear learner	Predict a numeric value
Recommendation	Factorization machines	Recommendations and time-series prediction
Unsupervised	K-means, PCA	Customer groupings; dimensionality reduction
Specialized	Image classification, Seq2Seq, NTM, LDA	Images and other deep learning tasks

06 TWO WAYS TO DEPLOY A MODEL

single predictions vs. a whole dataset

SAGEMAKER HOSTING SERVICES

persistent, real-time endpoint

- For single, on-demand predictions
- Compute instances behind a load-balanced HTTPS endpoint
- Applications call the endpoint API for inference
- Scale instances with demand; single- or multi-model endpoint

BATCH TRANSFORM

predictions for an entire dataset

- No permanent endpoint to maintain
- SageMaker spins up the model and predicts the whole dataset
- Stores results in Amazon S3, then terminates the instances
- Ideal for running a full validation set at test time

07 EXAM TRAPS

where the knowledge check gets you

- × "Accuracy is the best classification metric." — It misleads when **true negatives dominate** (fraud, disease): a high score can hide a poor ability to catch the rare positive. Use precision, sensitivity, or F1.
- × "You can evaluate on the training data." — That causes **overfitting** — the model memorizes rather than generalizes. Always hold out data the model has not seen (holdout or k-fold).
- × "Sensitivity and specificity are the same idea." — **Sensitivity** is the share of *positives* caught (recall/TPR); **specificity** is the share of *negatives* caught (TNR). A model can be high on one and low on the other.
- × "For a SageMaker CSV, put the target last and keep the header." — Backwards: SageMaker CSV files must have **no header row** and the **target in the first column**, features to its right.
- × "Assign 1, 2, 3… to unordered categories." — That invents a false **ordinal** relationship (the model thinks 2 outranks 1). Split **non-ordinal** data into one column per category instead.
- × "Hyperparameter tuning always improves the model." — The deck warns it *doesn't necessarily* help. Limit the hyperparameters and ranges you tune, and run **one training job at a time** for the best results.
- × "Deploy the model once and you're done." — Deployment is **continuous**: monitor production data and **retrain** as new patterns appear, because new data reveals new outcomes over time.
- × "RAG-style imputing: just drop every row with a missing value." — First learn *why* data is missing and how much: impute when values are few and missing at random; drop only when a whole row or column is largely empty.

08 SELF-QUIZ

answers are upside-down below

Q1 Most supervised ML business problems fall into which two categories?

Q2 An observation in labeled data is made up of which two elements?

Q3 What three-step process brings data from many systems into one repository?

Q4 Which fully managed, serverless AWS service runs ETL jobs using crawlers and a Data Catalog?

Q5 For a SageMaker CSV training file, where must the target column go and what must the file omit?

Q6 Which validation method splits a small dataset into K rotating segments so all data is used?

Q7 Evaluating a model on the same data it trained on causes what problem?

Q8 Which confusion-matrix metric is the proportion of positive predictions that are actually correct?

Q9 Which single metric combines precision and sensitivity?

Q10 Which SageMaker service analyzes data, auto-selects features, and trains/tunes models to find the best one?

ANSWER KEY (rotate the page)

Q1 classification and regression **Q2** the target and the features **Q3** ETL — extract, transform, load **Q4** AWS Glue **Q5** target in the first column; no header row **Q6** k-fold cross validation **Q7** overfitting **Q8** precision — TP ÷ (TP + FP) **Q9** the F1 score **Q10** Amazon SageMaker Autopilot