

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

COURSE SYLLABUS

SL 4M



COURSE SYLLABUS — SL 4M: MACHINE LEARNING ENGINEER

Maven Smart System (MSS) — USAREUR-AF

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

26 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

COURSE SYLLABUS — SL 4M: MACHINE LEARNING ENGINEER

MAVEN SMART SYSTEM (MSS) — USAREUR-AF

Field	Detail
Level	SL 4M (ML Engineer Specialist Track)
Duration	5 days (40 hours)
Prerequisites	SL 1, SL 2, SL 3 (all Go evaluations on file — REQUIRED , not recommended); Data Literacy Technical Reference (recommended); working Python proficiency (pandas, scikit-learn, PyTorch or equivalent); statistics fundamentals
Audience	ML engineers, data scientists building and deploying models on MSS
Format	Instructor-led lab + guided practice + practical exercise
Location	MSS Training Environment (GPU-enabled Code Workspace required)

BLUF: SL 4M teaches ML engineers to build a complete ML pipeline on MSS — from raw Foundry dataset through feature engineering, model training, evaluation, deployment, and production monitoring. Course focuses on operational implementation inside Foundry with the governance, monitoring, and documentation standards the Army requires. It does not teach ML theory; it assumes you can already train a model.

LEARNING OBJECTIVES

#	Objective
1	Configure a GPU-enabled Code Workspace for ML development with package management, environment reproducibility, and Foundry dataset connectivity
2	Build a feature engineering pipeline for a Foundry dataset meeting documented feature standards — null handling, encoding, scaling, leakage detection
3	Train and evaluate a supervised model against defined acceptance thresholds (accuracy, precision/recall, calibration, ROC-AUC)
4	Deploy a trained model to a Foundry model serving endpoint; verify live inference on test records
5	Implement a model monitoring pipeline with data drift detection and alert routing
6	Manage model versioning, experiment tracking, and the Foundry model registry
7	Connect a deployed model to Ontology Objects via Actions for operational integration
8	Complete a model governance document (model card) meeting USAREUR-AF documentation standards

PRE-COURSE CHECKLIST

Complete **10+ duty days before Day 1**:

- Request **GPU-enabled Code Workspace** from C2DAO — CPU-only workspaces are insufficient for model training exercises; allow 10+ duty days
- Confirm GPU allocation is active by running a provided test script before Day 1 — if it fails, contact C2DAO before arriving
- Read TM-40M, Chapter 1 (Introduction, role, prerequisites self-check) — 25 min
- Read TM-40M, Chapter 9 (Model Governance) — read this first; understanding the governance destination makes the technical content purposeful

DAILY SCHEDULE

Day 1 — Workspace Setup and Feature Engineering Fundamentals

Time	Block	Method	Content
0800–0900	1	Brief	MLE role on MSS; model governance overview; responsible AI principles for operational models; the full pipeline you will build this week
0900–1100	2	Lab	Code Workspace setup: GPU allocation, package management, Foundry dataset connectivity, session reproducibility
1100–1115	—	Break	
1115–1200	3	Lab	Foundry write pattern for ML: transaction-based output writes from Code Workspace — test this on Day 1, not Day 4
1200–1300	—	Lunch	
1300–1500	4	Lecture	Feature engineering principles: null handling strategies, encoding, scaling, leakage detection, feature documentation standards
1500–1515	—	Break	
1515–1700	5	Lab	Feature engineering practice: apply null handling, encoding, and leakage check to a provided dataset; document each feature

Evening reading: TM-40M, Chapter 3 (Feature Engineering — documentation standards).

Day 2 — Feature Engineering Pipeline Build

Time	Block	Method	Content
0800–0830	—	Review	Day 1 questions; leakage check methodology review
0830–1100	6	Lab	Feature pipeline build: raw Foundry dataset → complete feature matrix — null imputation, categorical encoding, numerical scaling, feature documentation
1100–1115	—	Break	

Time	Block	Method	Content
1115–1200	7	Lab	Feature pipeline: leakage audit — verify no feature can be derived from the label
1200–1300	—	Lunch	
1300–1500	8	Lab	Feature pipeline: output to Foundry curated dataset via write transaction; verify output schema and row counts
1500–1515	—	Break	
1515–1700	9	Lab	Experiment setup: train/test split, cross-validation configuration, baseline model (predict majority class) — establish the floor before building

Evening reading: TM-40M, Chapter 5 (Model Evaluation — acceptance thresholds and calibration requirements).

Day 3 — Model Training and Evaluation

Time	Block	Method	Content
0800–0830	—	Review	Day 2 questions; feature pipeline review — confirm each feature is documented and leakage-checked
0830–1100	10	Lab	Model training: scikit-learn and PyTorch patterns inside Code Workspace; cross-validation; hyperparameter tuning
1100–1115	—	Break	
1115–1200	11	Lab	Model evaluation: accuracy, precision/recall, ROC-AUC — applying USAREUR-AF acceptance thresholds; calibration check (Platt scaling / isotonic regression)
1200–1300	—	Lunch	
1300–1500	12	Lab	Model comparison exercise: train and evaluate two competing models; select winner; document decision rationale and trade-offs
1500–1515	—	Break	
1515–1700	13	Lab	Experiment tracking: logging experiment parameters and metrics to Foundry model registry; versioning models

Evening reading: TM-40M, Chapter 7 (MLOps — drift detection patterns and alert configuration).

Day 4 — Deployment and MLOps

Time	Block	Method	Content
0800–0830	—	Review	Day 3 questions; model evaluation standards review — calibration and its operational meaning
0830–1030	14	Lab	Model deployment: serving endpoint configuration, inference API setup, latency and throughput verification
1030–1045	—	Break	
1045–1200	15	Lab	Connecting deployed model to Ontology: Actions that invoke model inference, write predictions to Object properties
1200–1300	—	Lunch	
1300–1500	16	Lab	Monitoring pipeline: data drift detection — defining thresholds from validation set baseline (PSI, KS test); alert routing configuration
1500–1515	—	Break	
1515–1700	17	Lab	Operational use case patterns: readiness prediction, logistics demand forecasting, anomaly detection — patterns on MSS data with Ontology integration

Evening reading: TM-40M, Chapter 9 (Model Governance — re-read model card requirements; draft your governance document before arriving on Day 5).

Day 5 — Governance and Practical Exercise

Time	Block	Method	Content
0800–0900	18	Lab	Model governance: model card completion — model assumptions, training data description, known limitations, intended use restrictions, responsible AI declaration
0900–1000	19	Brief	Deployment approval process; C2DAO governance for deployed models; practical exercise scenario brief

Time	Block	Method	Content
1000–1015	—	Break	
1015–1100	20	Brief	Evaluation criteria review; planning time
1100–1200	—	Buffer	Questions / environment check
1200–1300	—	Lunch	
1300–1700	21	Eval	Practical exercise: feature pipeline → train → evaluate → deploy → monitoring → governance doc

REQUIRED READING SUMMARY

When	Reading
Before Day 1	TM-40M, Ch 1 (Introduction/prereq self-check)
Before Day 1	TM-40M, Ch 9 (Model Governance) — read first
Day 1 evening	TM-40M, Ch 3 (Feature Engineering — documentation standards)
Day 2 evening	TM-40M, Ch 5 (Model Evaluation — thresholds/calibration)
Day 3 evening	TM-40M, Ch 7 (MLOps — drift detection/alerting)
Day 4 evening	TM-40M, Ch 9 (re-read model card requirements; draft governance doc)

PRACTICAL EXERCISE

Scenario: The G4 needs a predictive model for equipment failure within the next 30 days, to prioritize pre-deployment maintenance. Provided: a Foundry dataset.

#	Task
1	Configure Code Workspace; verify connectivity and write transaction to the provided dataset

#	Task
2	Build the feature engineering pipeline: handle nulls, encode categoricals, document each feature, run leakage audit
3	Train a binary classifier for 30-day failure prediction; achieve $\geq 75\%$ recall (minimizing missed failures is the operational priority); calibrate the model
4	Produce an evaluation metrics report; document model assumptions and limitations
5	Deploy the model to a serving endpoint; verify inference on 10 provided test records
6	Implement a drift detection pipeline that alerts when input feature distributions shift beyond defined thresholds — the evaluator will seed a drift event; your pipeline must detect it
7	Complete the model governance document (model card) meeting TM-40M Chapter 9 standards

Go standard: Pass 6 of 7 tasks. Model meets minimum recall threshold. Drift detection pipeline catches the evaluator-seeded drift event. Governance document reviewed by technical evaluator.

GO CRITERIA

Task	Hard Standard
Governance document	A scored deliverable — missing model assumptions, training data description, known limitations, or intended use restrictions fails that element. Write it as if a commander's analyst will use it to decide whether to trust the model's outputs
Drift detection	Must catch the evaluator-seeded drift event within the defined threshold — if it does not, that task is No-Go
Model calibration	If you say "85% probability of failure," the model should produce a failure 85% of the time at that score — uncalibrated probabilities on operational prediction products fail the evaluation

KEY TIPS

Risk	Guidance
Code Workspace session state	Persists between sessions but session state does not always — start each session by re-running setup cells. Build a clean setup block you can re-run in 30 seconds on Day 1

Risk	Guidance
Feature leakage	Most common pipeline failure. Run a leakage check before the evaluation: can any feature be derived from the label?
Model calibration	Ignored in most academic ML work — Army operational products require calibrated probabilities. Use Platt scaling or isotonic regression. Chapter 5 has the procedure
Drift detection threshold	Must be a real threshold — not "any change." Define it based on validation set baseline and document why. "Alert when PSI > 0.2" is defensible
Foundry write transaction	Test this on Day 1 (Block 3), not Day 4. Trainees who skip Block 3 consistently lose time during the evaluation when the write fails

CONTINUATION

Graduates managing production ML systems or building platform infrastructure used by other data scientists may pursue **SL 5M (Advanced ML Engineering)**. SL 5M covers enterprise drift detection, automated retraining pipelines, feature stores, ensemble architecture at scale, model governance documentation, and production model audits. Prerequisites: SL 4M Go evaluation on file; 12+ months active ML engineering experience in a production context.

ASSOCIATED EXERCISES AND ASSESSMENTS

Item	Reference
Pre-course exam	EXAM_TM40M_PRE
Post-course exam	EXAM_TM40M_POST
Practical exercise	EX_40M (EXERCISE.md + ENVIRONMENT_SETUP.md)

RELATIONSHIP TO WFF TRACKS

WFF track analysts (SL 4A through SL 4F) are the operational consumers of predictive models built in this course. ML engineers should understand the prediction products each WFF community uses: sustainment analysts (SL 4D) consume equipment failure prediction and demand forecasting models;

intelligence analysts (SL 4A) use anomaly detection outputs; protection analysts (SL 4E) rely on readiness prediction models for force protection resourcing.

USAREUR-AF Operational Data Team Syllabus SL 4M | Version 2.0 | March 2026

DRAFT