JUPITER PRIME

Consulting • Training • Advisory

جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

Advanced Machine Learning Engineering and Experimentation Training Course

jupiterprime.uk

info@jupiterprime.uk

Registered in the United Kingdom and Saudi Arabia

Advanced Machine Learning Engineering and Experimentation Training Course

Advanced Machine Learning Engineering and Experimentation is a 2-week training course for data scientists, ML engineers, AI researchers and technical leads that ends with a research-grade ML experiment report on one model built and validated during the course. As teams move from fine-tuning off-the-shelf models to designing their own architectures, weak experimental practice produces results that cannot be reproduced or trusted. Jupiter Prime Training delivers this Advanced Machine Learning Engineering and Experimentation course.

Why Machine Learning Engineering and Experimentation Matters Now

Three pressures shape machine learning engineering and experimentation today: transformer and diffusion architectures that change faster than in-house knowledge, compute budgets that punish unplanned hyperparameter searches, and reviewers who reject claims built on leaked test sets or single random seeds. A group that runs disciplined experiments ships models whose gains survive production traffic; one that does not chases benchmark noise and burns GPU hours. The course links engineering practice to ISO/IEC 42001, the AI management system standard, and the NIST AI Risk Management Framework NIST AI RMF for documenting validity and reliability.

Who Gains Most from Machine Learning Engineering and Experimentation

- Specialists who design, train and benchmark deep learning architectures
- Engineers who own training pipelines, distributed compute and experiment tracking
- Researchers who publish internal or external findings and defend them under review
- Technical leads who approve which model variant moves to production
- Applied scientists who build reinforcement learning or probabilistic forecasting systems

An applied research or ML platform group gains most when several of its engineers and scientists attend together and adopt one protocol for experiment design and reporting.

Machine Learning Engineering and Experimentation Capabilities

Participants move from architecture choice to statistically sound comparison, ending with results that another engineer can reproduce from the report alone.

- Design controlled experiments with ablations, seeds and pre-registered hypotheses — research leads and reviewers
- Build and tune transformer, convolutional and graph architectures — model owners
- Apply Bayesian and probabilistic methods to quantify predictive uncertainty — risk-sensitive product teams
- Test reinforcement learning agents for stability and reward hacking — safety and operations owners
- Calculate statistical significance and confidence intervals for model comparisons — technical leads
- Specify reproducible training pipelines with versioned data, code and configuration — ML platform teams

Key Machine Learning Engineering and Experimentation Terms

- Ablation study — removing or changing one component at a time to measure its contribution to model performance.
- Self-attention — the transformer mechanism that weights every input token against every other to build context-aware representations.
- Hyperparameter optimisation — systematic search over settings such as learning rate and depth using grid, random or Bayesian methods.
- Data leakage — information from the test set or the future entering training, inflating reported accuracy.
- Calibration — how closely a model's predicted probabilities match the observed frequency of outcomes.
- Policy gradient — a reinforcement learning method that adjusts an agent's action probabilities directly to increase expected reward.

Machine Learning Engineering and Experimentation Day by Day

Day 1: Research Questions and Experimental Design

- Hypothesis Framing and Success Metrics — What Counts as an Improvement
- Baselines and Strong Comparators — Avoiding Inflated Gains
- Train, Validation and Test Protocols — Preventing Data Leakage
- Seeds, Variance and Repeated Runs — Separating Signal From Noise
- Experiment Tracking and Versioning — Reproducibility From Day One

Day 2: Optimisation and Training Dynamics

- Loss Landscapes and Optimiser Choice — Adam, SGD and Schedulers
- Initialisation and Normalisation Layers — Stable Deep Network Training
- Regularisation Methods — Dropout, Weight Decay and Early Stopping
- Gradient Diagnostics — Vanishing, Exploding and Clipping Decisions
- Mixed Precision and Distributed Training — Compute Budget Trade-Offs

Day 3: Convolutional and Graph Architectures

- Residual and Efficient Convolutional Networks — Depth Versus Latency
- Graph Neural Networks — Learning From Relational and Network Records
- Transfer Learning and Fine-Tuning Strategies — Frozen Versus Full Updates
- Data Augmentation Policies — Robustness Without Label Noise
- Architecture Search Heuristics — Where Manual Design Still Wins

Day 4: Transformers and Sequence Modelling

- Self-Attention and Positional Encoding — Inside the Transformer Block
- Encoder, Decoder and Encoder-Decoder Variants — Task Fit
- Parameter-Efficient Fine-Tuning — Adapters and Low-Rank Updates
- Long-Context and Efficient Attention — Memory and Throughput Limits
- Time-Series Transformers — Forecasting in Energy and Retail Demand

Day 5: Probabilistic and Bayesian Modelling

- Bayesian Inference Foundations — Priors, Posteriors and Evidence
- Variational Inference and Monte Carlo Dropout — Scalable Uncertainty
- Gaussian Processes — Small-Sample Regression With Confidence Bands
- Calibration Curves and Temperature Scaling — Trustworthy Probabilities
- Probabilistic Graphical Models — Encoding Causal Assumptions

Day 6: Reinforcement Learning Engineering

- Markov Decision Processes and Value Functions — Problem Formulation
- Policy Gradient and Actor-Critic Methods — Choosing an Algorithm
- Reward Design and Reward Hacking — Aligning Agent Incentives
- Offline Reinforcement Learning — Learning From Logged Decisions
- Simulation Environments — Logistics Routing and Grid Control Cases

Day 7: Generative Architectures

- Variational Autoencoders — Latent Spaces and Reconstruction Quality
- Diffusion Models — Noise Schedules and Sampling Steps
- Generative Model Evaluation Metrics — Fidelity Versus Diversity
- Synthetic Data Generation — Privacy and Distribution Shift Risks
- Instruction Tuning and Preference Optimisation — Shaping Model Behaviour

Day 8: Evaluation Methods and Statistical Rigour

- Cross-Validation Schemes — Grouped, Stratified and Time-Aware Splits
- Significance Testing and Bootstrap Intervals — Comparing Two Models
- Robustness and Out-of-Distribution Tests — Behaviour Under Shift
- Fairness Metrics Across Subgroups — Disparity Detection
- NIST AI RMF Measure Function — Documenting Validity and Reliability

Day 9: Hyperparameter Search and Reproducible Pipelines

- Bayesian Optimisation and Successive Halving — Efficient Search Budgets
- Configuration Management — Versioned Code, Data and Environments
- Containerised Training Jobs — Identical Runs Across Clusters
- Model Cards and Datasheets — Recording Intended Use and Limits
- ISO/IEC 42001 Records — Linking Experiments to Management Controls

Day 10: Practising the ML Experiment Report

- Hypothesis and Protocol From Day 1 — Report Methods Section
- Architecture and Tuning Results From Days 3 to 7 — Findings
- Statistical Evidence From Day 8 — Significance and Uncertainty
- Reproducibility Package From Day 9 — Code, Data and Configurations
- Research-Grade ML Experiment Report — Peer Review Presentation

Applying Machine Learning Engineering and Experimentation After the Course

- Re-run one recent model comparison with five seeds and bootstrap intervals, and record whether the claimed improvement still holds.
- Introduce an experiment template with hypothesis, baselines, splits and ablation plan for every new modelling ticket in the next sprint.

Machine Learning Engineering and Experimentation Questions FAQ

What is advanced machine learning engineering and experimentation and why does it matter now?

Advanced machine learning engineering and experimentation is the disciplined design, training and testing of deep, probabilistic and reinforcement learning models. It matters now because architectures change quickly and compute is costly, so only rigorous experiments show which gains are real.

Which roles gain most from advanced machine learning engineering and experimentation training?

Data scientists, ML engineers, AI researchers and technical leads gain most. Applied scientists building forecasting or decision agents also benefit, since uncertainty and reinforcement learning methods are covered in engineering depth.

How does advanced machine learning engineering and experimentation differ from a general AI and machine learning course for professionals?

It assumes coding and modelling experience and works at research depth. You design ablations, tune transformers, quantify uncertainty and test significance, rather than surveying techniques and use cases at an overview level.

What do you bring back from advanced machine learning engineering and experimentation training?

You bring back a research-grade ML experiment report on one model. It states the hypothesis, protocol, results, statistical evidence and a reproducibility package that colleagues can rerun.

Machine Learning Engineering and Experimentation: The Way Forward

The research-grade ML experiment report informs the first technical decision: whether the new model variant earns promotion over the current baseline. Technical leads, researchers and platform engineers use it in review. After the model meets live traffic, revisit calibration and out-of-distribution results, since production inputs often expose gaps that offline benchmarks hid.