ParhoAI

PARHOAI.COM

3 MONTHS

AI Engineering Bootcamp

For engineers who already code. Thirteen weeks from your first production model to a deployed agentic system.

3 months	52	78	6	5 + 1
DURATION	SESSIONS	CONTACT HOURS	MODULES	PROJECTS & CAPSTONE

The bootcamp at a glance

Three months. Four sessions a week, ninety minutes each, across thirteen weeks — **52 sessions and 78 contact hours**, plus eight to ten hours of self-study. This is a conversion course, not an introduction: it assumes you can already program and have trained a model.

	MODULE	SESSIONS	PROJECT
01	**Production-grade ML**	6	Project 1 — A model that would survive review
02	**Deep learning**	8	Project 2 — A trained network, from scratch
03	**Computer Vision**	10	Project 3 — A vision system on live video
04	**NLP and Transformers**	7	Project 4 — A fine-tuned language model
05	**Agentic AI and LLMs**	12	Project 5 — An agentic application
06	**Production and capstone**	9	Capstone — Deployed, documented, defended

PREREQUISITES — THIS COURSE DOES NOT TEACH THESE

Programming — Python at working level — functions, classes, comprehensions, virtual environments, packaging. You write code most days.

Data — SQL joins and aggregates. pandas for loading, cleaning and reshaping a dataframe.

ML fundamentals — Supervised vs unsupervised. Train/test split and why. What overfitting is. You have trained at least one scikit-learn model and read a confusion matrix.

Tooling — Git and GitHub: branches, merges, pull requests. Comfortable in a Linux or macOS terminal. Docker is helpful, not required.

Web — HTTP verbs and status codes, REST, JSON. You have called and ideally built an API.

Experience — You have built and maintained something real. This course assumes engineering judgement, not just syntax.

Entry is assessed. A short take-home and a fifteen-minute call before a place is confirmed. Admitting someone who does not meet the bar wastes their money and slows the cohort — if you are not there yet, the eight-month diploma is the right route.

ASSESSMENT **50%** module projects **30%** capstone **10%** code review **10%** weekly exercises

No written exams

MODULE
01 Production-grade ML

You know how to train a model. This is what separates one that ships.

6 SESSIONS
9 HOURS
1 – 6

ASSUMED GOING IN

Python, pandas and scikit-learn at working level · you have trained and evaluated at least one supervised model · you know what a train/test split and a confusion matrix are

This module does not explain what machine learning is. It assumes you have built a model and now need one that survives production.

001 **From notebook to production ML**

What breaks when a model leaves the notebook · assumptions that hold in training and fail in deployment · the model is 10% of the system

002 **Evaluation beyond accuracy**

Cost-sensitive thresholds · PR-AUC vs ROC-AUC under imbalance · probability calibration · decision curves · picking the metric from the business, not the paper

003 **Validation design and leakage audits**

Temporal, group and nested cross-validation · backtesting · auditing a pipeline for target leakage · why your 0.98 is a bug

004 **Boosting at a professional level**

XGBoost, LightGBM, CatBoost compared · Optuna · native categorical handling · early stopping · when to stop tuning

005 **Interpretability and trust**

SHAP · permutation importance · partial dependence · counterfactuals · explaining a model to someone who will not read the notebook

006 **Feature engineering + LAB**

Target encoding without leakage · interactions · feature stores · leakage-safe pipelines · supervised build clinic

PROJECT 1

A model that would survive review

A real problem on real data, evaluated the way a business would judge it — not by accuracy.

DELIVERABLE

Leakage-audited pipeline · cost-based threshold · SHAP explanations · stated limitations

PROVES

Validation design · cost-sensitive evaluation · interpretability

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Credit default risk — public lending data, cost-weighted thresholds, fairness analysis across groups
- Subscription churn — cost of a false positive vs a missed churner, modelled explicitly
- Retail demand forecasting — SKU-level, with seasonality and promotion effects
- Insurance claim fraud — severe class imbalance, precision at fixed recall, reviewer workload

MODULE
02 Deep learning

Neural networks, without the magic.

8 SESSIONS
12 HOURS
7 – 14

ASSUMED GOING IN

Module 1 or equivalent · NumPy and vectorization · linear algebra: matrices and dot products · calculus: partial derivatives and the chain rule · comfortable reading mathematical notation

No prior PyTorch required. Prior exposure to any framework helps but is not assumed.

007 **Networks and backpropagation**

Depth and representation · activations · initialization · vanishing and exploding gradients · when a network is the wrong choice

008 **PyTorch**

Tensors, devices, autograd · nn.Module · writing a training loop from scratch · checkpointing and resuming

009 **Losses, optimizers and schedules**

Cross-entropy, focal, custom losses · SGD, Adam, AdamW · learning-rate finding · warmup and cosine decay

010 **Regularization and normalization**

Dropout · weight decay · batch, layer and group norm · early stopping · what actually reduces overfitting in practice

011 **Data pipelines**

Dataset and DataLoader · transforms and augmentation · samplers for imbalance · workers, prefetching, and I/O bottlenecks

012 **Training at scale**

GPU memory accounting · batch size · mixed precision · gradient accumulation · multi-GPU in principle · Colab and cloud GPUs

013 **Debugging training runs**

Loss will not move · NaNs · silent label bugs · overfitting a single batch as a sanity check · gradient inspection

014 **Experiment discipline + LAB**

Seeds and determinism · baselines before models · ablations · tracking runs · supervised build clinic

PROJECT 2

A trained network, from scratch

Take a task, build the dataset, train a network, and show the experiment log.

DELIVERABLE

Training curves · baseline comparison · one documented ablation · reproducible seed

PROVES

PyTorch · training loops · experiment discipline

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Urdu or Arabic handwritten character recognition, with your own collected dataset
- Machine fault detection from audio — spectrograms, industrial recordings
- Document image restoration — denoising and deskewing scanned records
- Deep learning vs gradient boosting on tabular data — benchmarked honestly, with the loser reported

MODULE
03 Computer Vision

Detection, video, and the edge.

10 SESSIONS
15 HOURS
15 – 24

ASSUMED GOING IN

Module 2 or equivalent PyTorch experience · you can write, run and debug a training loop unaided · basic image handling · GPU access, local or cloud

This module does not teach deep learning from scratch. It applies it to vision problems at production scale.

015 **Images as tensors**

Pixels, channels, color spaces · resizing and normalization · OpenCV · classical CV where it still beats deep learning

016 **Convolutions and CNN architectures**

Kernels, stride, padding, receptive field · ResNet and skip connections · EfficientNet · Vision Transformers in brief

017 **Transfer learning**

Pretrained backbones · freezing and unfreezing · fine-tuning schedules · what to do when you have 200 images

018 **Data collection and labeling**

Sourcing and class balance · CVAT, Roboflow, Label Studio · label quality and inter-annotator agreement · dataset versioning

019 **Object detection**

Bounding boxes · IoU · non-max suppression · anchor vs anchor-free · mAP explained properly

020 **YOLO in practice**

Architecture overview · Ultralytics · inference · confidence and NMS thresholds · latency vs accuracy

021 **Training a custom detector**

Dataset formats · augmentation for detection · hyperparameters · evaluating on your own held-out data

022 **Segmentation, pose and keypoints**

Semantic vs instance segmentation · U-Net · Segment Anything · human pose · masks, IoU and Dice

023 **Video, tracking and live streams**

Frames vs streams · RTSP and ONVIF · buffering, dropped frames, latency budgets · ByteTrack · re-identification

024 **Edge deployment + LAB**

ONNX · TensorRT · quantization and pruning · CPU and Jetson inference · FPS benchmarking · supervised build clinic

PROJECT 3

A vision system on live video

Collect and label your own data, train a detector, run it on a live or recorded stream.

DELIVERABLE

Custom-trained model · runs on video · measured mAP and FPS · demo clip with boxes drawn

PROVES

Transfer learning · YOLO · tracking · edge inference

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- PPE compliance on construction footage — helmet and vest detection, per-zone rules, violation log
- Number plate recognition for a car park — detection, OCR, entry and exit reconciliation
- Retail shelf monitoring — out-of-stock and planogram compliance from a fixed camera
- Queue and dwell analytics — a clinic, bank or store, with wait-time estimation
- Traffic analytics — vehicle counting, classification and wrong-way detection on road video

MODULE
04 NLP and Transformers

Text, embeddings, fine-tuning.

7 SESSIONS
10.5 HOURS
25 – 31

ASSUMED GOING IN

Module 2 or equivalent PyTorch experience · training loops and fine-tuning as a concept · tokenization is introduced, not assumed · GPU access

Independent of Module 3 — computer vision is not a prerequisite for this one.

025 **Tokenization and embeddings**

BPE and WordPiece · vocabularies · word2vec to sentence transformers · similarity · Unicode and Urdu pitfalls

026 **Attention and the transformer**

Queries, keys, values · multi-head attention · positional encoding · encoder vs decoder · BERT and GPT families

027 **The Hugging Face ecosystem**

Model hub · tokenizers · pipelines · datasets · reading a model card properly · licensing traps

028 **Fine-tuning**

Task heads · training arguments · freezing layers · catastrophic forgetting · overfitting on small datasets

029 **LoRA and QLoRA**

Parameter-efficient tuning · adapters · rank and alpha · merging weights · when full fine-tuning is unnecessary

030 **Evaluating NLP systems**

F1 and accuracy · BLEU and ROUGE · perplexity · human evaluation · designing a test set that survives contact

031 **NLP build clinic** LAB

Supervised build and evaluation of your own fine-tuned model

PROJECT 4

A fine-tuned language model

Fine-tune a small model for a specific task. Urdu or bilingual work strongly encouraged.

DELIVERABLE

Fine-tuned checkpoint · evaluation against a baseline · error analysis · model card

PROVES

Transformers · Hugging Face · LoRA · evaluation

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Support ticket triage — Urdu and English, intent classification plus urgency scoring
- Contract and legal clause extraction — NER over real document types, with confidence
- Resume parser and role matcher — structured extraction plus a defensible relevance score
- Roman Urdu moderation — normalization, abuse detection, and an honest false-positive analysis

MODULE
05 Agentic AI and LLMs

RAG, tools, agents, guardrails.

12 SESSIONS
18 HOURS
32 – 43

ASSUMED GOING IN

Python, async and REST APIs · JSON and schema thinking · you have built and consumed an API · basic ML literacy

Deep learning is NOT a prerequisite for this module. It is the most self-contained module in the bootcamp.

032 **How LLMs work**

Pretraining and next-token prediction · instruction tuning · RLHF and DPO · what emerges, what does not, and what they will confidently invent

033 **Choosing a model**

Hosted vs open weights · size vs cost vs latency · licensing · reading benchmarks skeptically · when a small model is enough

034 **Prompting as engineering**

System prompts · few-shot examples · chain of thought · decomposition · prompt versioning and diffing

035 **Structured output**

JSON mode · schemas · Pydantic validation · retries on malformed output · building type-safe LLM pipelines

036 **Embeddings and vector stores**

Embedding model choice · similarity metrics · pgvector, Qdrant, Chroma · index types · recall vs latency

037 **RAG I – ingestion**

Document parsing including PDFs and tables · chunking strategies · overlap · metadata design · incremental updates

038 **RAG II – retrieval**

Dense and hybrid retrieval · BM25 · reranking · query rewriting and expansion · returning citations users can check

039 **RAG III – failure modes**

Hallucination · retrieval misses · stale data · chunk boundary problems · measuring retrieval quality separately from generation

040 **Evaluating LLM systems**

Golden sets · LLM-as-judge and its limits · regression testing prompts · human review loops · what to log

041 **Tool calling and agents**

Tool schemas · execution and error handling · sandboxing · ReAct · planning · loop control and stopping conditions

042 **Memory, multi-agent and orchestration**

Short vs long-term memory · conversation state · summarization · supervisor patterns · hand-offs · when it is plainly overkill

043 **Guardrails and safety + LAB**

Prompt injection · jailbreaks · PII handling · content filtering · human in the loop · supervised build clinic

PROJECT 5

An agentic application

An LLM system that retrieves, uses tools, and does something a person would pay for.

DELIVERABLE

Working app · RAG over a real corpus · tool calling · written evaluation · cost per query

PROVES

RAG · vector stores · tool calling · agents · evaluation · guardrails

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Internal knowledge assistant — RAG over real company documents, with citations and access control
- Support agent with escalation — resolves what it can, hands off cleanly with full context
- Invoice and purchase-order processing — extract, validate against rules, flag exceptions for a human
- SQL analyst agent — natural language to query over a real schema, with read-only guardrails
- Research agent — gathers from multiple sources, cross-checks, cites, and states uncertainty

MODULE

06 Production and capstone

MLOps, LLMOps, and shipping the thing.

9 SESSIONS

13.5 HOURS

44 – 52

ASSUMED GOING IN

Git and the command line · a trained model of your own to deploy, from an earlier module · Docker familiarity helps but is taught from first principles · a cloud account

This module assumes you have something worth deploying. It is not a DevOps introduction.

044 **Docker and packaging**

Images and containers · Dockerfile · multi-stage builds · slim and GPU images · registries · reproducible environments

045 **Serving models**

FastAPI plus a model · batch vs real-time · ONNX Runtime · concurrency and workers · load testing before launch

046 **CI/CD and cloud**

GitHub Actions · test, build and deploy pipelines · environments and secrets · IAM and least privilege · controlling cost

047 **Experiment tracking and registry**

MLflow and Weights & Biases · runs, params, artifacts · model registry · staging and promotion · rolling back a bad model

048 **Monitoring and drift**

Metrics, logs and traces · data drift vs concept drift · alerting that fires · Prometheus and Grafana · shadow deployment

049 **LLMOps**

Why MLOps habits do not transfer · non-determinism · tracing with Langfuse or LangSmith · prompt version control · online evaluation

050 **Cost, caching and reliability**

Semantic caching · model routing · fallbacks and graceful degradation · rate limits and outages · cost per request as an SLO

051 **capstone build** LAB

Supervised build time with instructor review

052 **Demo day** DEMO

Present the capstone · live demo · questions from a panel and invited guests

CAPSTONE

Deployed, documented, defended

A system of your choosing that solves a real problem for a real user.

DELIVERABLE

Live URL or reproducible deployment · monitoring · README with known limitations · presented and questioned

PROVES

Everything above

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Deploy your Module 6 detector as a monitored service — drift alerts, latency SLO, rollback path
- Multi-model serving with A/B routing and per-model cost and latency dashboards
- Automated retraining pipeline — triggered by drift, gated by an evaluation threshold
- GPU inference service with autoscaling and a hard monthly cost ceiling

FINAL

Capstone options

Industry problems, not exercises. Choose one, or bring your own of equal weight.

15 OPTIONS

5 SECTORS

WHAT MAKES A CAPSTONE INDUSTRY-GRADE

- A real user or a credible proxy for one — not a dataset alone.
- Real data, with the mess left in. Kaggle-clean data does not qualify.
- Deployed and reachable — a live URL or a reproducible deployment.
- Monitored, with logging and at least one alert that would actually fire.
- Evaluated against a stated baseline, with the metric justified in business terms.
- Documented limitations. What it gets wrong, and who should not rely on it.

SAFETY AND INDUSTRIAL

01 Site safety monitoring platform

Multi-camera PPE, restricted-zone and unattended-object detection, with an operator dashboard, alert routing and a published false-positive rate.

02 Traffic incident detection

Wrong-way, stopped-vehicle and congestion detection across public camera feeds, with time-to-alert measured against a manual baseline.

03 Predictive maintenance

Sensor time-series from industrial equipment, failure prediction with a costed lead-time trade-off, and alerting into an existing workflow.

RETAIL AND COMMERCE

04 Store analytics platform

Footfall, dwell, queue length and conversion by zone from existing CCTV, benchmarked against till data.

05 Dynamic pricing engine

Competitor scraping, demand elasticity modeling, and a pricing recommendation with guardrails and an audit trail.

06 Catalog enrichment at scale

Image classification plus LLM-generated descriptions and attributes for a large product catalog, with human review sampling.

FINANCE AND RISK

07 Credit scoring service

Deployed scoring API with SHAP explanations, audit logging, challenger models and a documented fairness assessment.

08 Real-time fraud detection

Streaming transaction scoring with sub-100ms latency, threshold tuning against reviewer capacity, and drift monitoring.

09 Document-driven KYC pipeline

ID extraction, verification checks, PII redaction and an exception queue for human review.

LANGUAGE AND KNOWLEDGE

10 Bilingual enterprise assistant

Urdu–English RAG over real internal documents, with citations, access control, and evaluation against a golden set.

11 Automated compliance checker

Reads policy documents, flags violating clauses in contracts, and cites the specific rule breached.

12 Multilingual support agent

Handles tier-one queries in Urdu and English, escalates with full context, and reports containment rate and cost per resolution.

HEALTH, AGRICULTURE AND PUBLIC SECTOR

13 Clinical document triage

De-identification, classification and routing of medical records, with a measured PII leakage rate.

14 Crop disease detection

Phone-camera classification with an offline mobile path, built for low bandwidth and imperfect images.

15 Transport crowding prediction

Forecast platform and vehicle crowding from camera and ticketing data, to inform scheduling decisions.