ParhoAI

PARHOAI.COM

8 MONTHS

Applied AI Engineering Diploma

From no computer-science background to a shipping AI engineer, with a portfolio that proves it.

8 months	140	210	10	9 + 1
DURATION	SESSIONS	CONTACT HOURS	MODULES	PROJECTS & CAPSTONE

The program at a glance

Eight months. Four sessions a week, ninety minutes each, across thirty-five weeks — **140 sessions and 210 contact hours**, plus six to eight hours of self-study a week. Every module ends with a project that ships to a public GitHub repository. The portfolio is the qualification; the certificate is a formality.

	MODULE	SESSIONS	PROJECT
01	Foundations	14	Project 1 — A command-line tool
02	Engineering	16	Project 2 — A tested, installable package
03	Data and the internet	16	Project 3 — A data pipeline that runs unattended
04	Machine learning	16	Project 4 — End-to-end machine learning
05	Deep learning	12	Project 5 — A trained network, from scratch
06	Computer Vision	16	Project 6 — A vision system on live video
07	NLP and Transformers	12	Project 7 — A fine-tuned language model
08	Agentic AI and LLMs	16	Project 8 — An agentic application
09	DevOps and MLOps	12	Project 9 — A deployed, monitored service
10	LLMOps and capstone	10	Capstone — Deployed, documented, defended

WHO THIS IS FOR

Anyone who can use a computer and read English technical text. **No prior coding required.** Module 1 starts at the terminal.

Instruction in Urdu. All material and code in English, because that is what the job is in.

WHAT IT TAKES

6 contact hours a week over 8 months, plus 6–8 hours of self-study. The self-study is not optional.

A student who does only the contact hours will finish and struggle to get hired. We say this before enrolment, not after.

WHAT YOU LEAVE WITH

Nine projects and a deployed capstone, all public, all defensible in an interview.

The ability to take a real problem, build a model, deploy it, monitor it, and say honestly what it cannot do.

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

No written exams

MODULE
01 Foundations

The computer stops being a mystery box.

14 SESSIONS

1 – 14

---

001 **How a computer runs a program**

CPU, memory and storage · compiled vs interpreted · what an operating system actually does · why engineers use Linux

002 **Installing Linux / WSL**

WSL2 vs dual boot vs VM · choosing a distribution · first boot · terminal emulator setup

003 **The filesystem**

Absolute and relative paths · /home /etc /var /tmp · hidden files · tree and disk usage

004 **Shell essentials**

ls cd cp mv rm mkdir touch cat less · tab completion · command history · man pages

005 **Text processing in the shell**

Pipes and redirects · grep · find · wc sort uniq head tail · xargs · chaining commands

006 **Users, permissions and processes**

chmod and chown · sudo · ps top kill · services · apt and package management

007 **Editor and environment**

VS Code setup · extensions · integrated terminal · dotfiles · generating and using SSH keys

008 **Git fundamentals**

Repositories · the staging area · commit, log, diff · restore · HEAD · writing a useful commit message

009 **Git branching**

branch and switch · merge · resolving conflicts · stash · a sane branching workflow

010 **GitHub**

Remotes · clone push pull · forks · pull requests · issues · code review etiquette

011 **Documentation and repo hygiene**

Markdown · README structure · .gitignore · licenses · a GitHub profile that reads well

012 **Python I — values and types**

The REPL · numbers, strings, booleans · variables · operators · f-strings · string methods

013 **Python II — control flow**

Conditionals and truthiness · for and while · range · break and continue · nesting

014

Python III — functions and errors

Definitions · args and kwargs · return · scope · exceptions · reading a traceback · reading documentation

PROJECT 1

A command-line tool

Something with genuine use — a file organizer, an expense tracker, a bulk renamer.

DELIVERABLE

Public GitHub repository · README a stranger can follow · clean commit history

PROVES

Shell · Git · Python fundamentals

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Server log triage tool — parse production logs, cluster error patterns, output a daily digest
- Repository auditor — scan a codebase for committed secrets, oversized files and stale dependencies
- Bulk asset pipeline — rename, resize and catalog product images for an e-commerce feed
- Ops report generator — turn raw CSV exports into a formatted weekly report with charts

MODULE
02 Engineering

Stop writing scripts. Start writing software.

16 SESSIONS

15 – 30

---

015 **Lists and tuples**

Indexing and slicing · mutation · unpacking · sorting with keys · when a tuple is the right choice

016 **Dictionaries and sets**

Keys and lookup cost · nested data · defaultdict and Counter · set operations · deduplication

017 **Comprehensions and iteration**

List, dict and set comprehensions · generators and yield · itertools · laziness and memory

018 **Files and formats**

open and with · encodings · JSON · CSV · pathlib · reading large files without exhausting memory

019 **Modules and project structure**

Imports and namespaces · packages · __init__.py · src layout · entry points · circular imports

020 **Environments and dependencies**

venv · pip · requirements vs pyproject · uv and poetry · pinning · reproducible installs

021 **Project 1 build clinic** LAB

Supervised build time · pair debugging · instructor walkthrough of common mistakes

022 **Project 1 review**

Refactor a script into a package · peer code review · commit hygiene · what reviewers look for

023 **OOP I – classes and objects**

Classes vs instances · attributes and methods · self · modeling state · a real domain object

024 **OOP II – making classes work**

__init__ · dunder methods · __repr__ · properties · dataclasses · equality and hashing

025 **OOP III – design**

Inheritance vs composition · duck typing · abstract base classes · when OOP is the wrong tool

026 **Testing with pytest**

Test structure · assertions · parametrize · coverage · what is worth testing and what is not

027 **Fixtures, mocking and TDD**

Fixtures and scope · monkeypatch · mocking I/O and APIs · red–green–refactor in practice

028 **Debugging, logging and quality**

breakpoint and pdb · the VS Code debugger · logging levels · type hints · ruff, black, pre-commit

029 **Complexity and data structures**

Big-O without the maths degree · arrays vs hash maps · string handling · choosing the right structure

030 **Algorithms in practice**

Recursion · stacks and queues · trees · graph traversal · sorting and searching · enough for interviews

PROJECT 2

A tested, installable package

A real utility library, packaged properly, with a test suite that catches regressions.

DELIVERABLE

pip-installable · pytest suite · type hints · CI green on GitHub Actions

PROVES

OOP · testing · packaging · continuous integration

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Typed SDK for a public API — government open data, exchange rates or a payments sandbox, with retries and backoff
- Data-validation library — declarative schemas for CSV and JSON imports, with useful error messages
- Pricing and tax engine — slabs, discounts and rounding rules, at 100% branch coverage
- Urdu–English text normalization package — script handling, transliteration, cleaning

MODULE
03 Data and the internet

Get data from anywhere to anywhere.

16 SESSIONS
31 – 46

031 **How the web works**
Client and server · HTTP verbs · status codes · headers · cookies and sessions · curl

032 **Networking essentials**
DNS · IP and ports · SSH · TLS and HTTPS · firewalls · what a reverse proxy does

033 **Consuming APIs**
REST conventions · authentication with keys, OAuth and JWT · pagination · the requests library

034 **Building robust clients**
Rate limits · retries with exponential backoff · timeouts · idempotency · handling partial failure

035 **Web scraping**
requests and BeautifulSoup · Playwright for dynamic pages · robots.txt · legality and ethics

036 **Building an API with FastAPI**
Routes · path and query parameters · Pydantic models · request and response bodies · status codes

037 **API quality**
Validation · auto-generated OpenAPI docs · dependency injection · testing endpoints · versioning

038 **Async and concurrency**
async and await · I/O-bound vs CPU-bound · threads vs processes · when async actually helps

039 **Database design**
Entities and relationships · normalization · primary and foreign keys · when to denormalize

040 **SQL I**
SELECT WHERE ORDER BY LIMIT · GROUP BY and aggregates · NULL handling · CASE

041 **SQL II**
Inner, left and right joins · self joins · subqueries · CTEs · window functions

042 **Performance and integrity**
Indexes and when they help · EXPLAIN · transactions and ACID · connection pooling

043 **PostgreSQL from Python**
psycopg · SQLAlchemy · migrations with Alembic · connection strings and secrets

044 **NumPy**

ndarrays and dtypes · broadcasting · vectorization · why Python loops are slow · random seeds

045 **pandas I**

Series and DataFrame · selection and filtering · dtypes · missing values · genuinely messy cleaning

046 **pandas II and visualization**

groupby · merge and join · pivot and reshape · time series · matplotlib and seaborn

PROJECT 3

A data pipeline that runs unattended

Pull from an API or scrape a site, clean it, store it, expose it, chart it.

DELIVERABLE

Scheduled job · Postgres schema · FastAPI endpoint · dashboard · survives failure

PROVES

HTTP · APIs · SQL · pandas · scheduling

INDUSTRY-GRADE OPTIONS — CHOOSE ONE

- Market data pipeline — PSX or crypto, daily ingest, corporate actions, backfill, dashboard
- Property listing aggregator — scrape, normalize, deduplicate, track price per square foot by area
- Competitor price tracker — scheduled scrape, change detection, alert on undercuts
- Air quality and weather warehouse — multi-source ingest, gap filling, threshold alerting

MODULE
04 Machine learning

The classical toolkit still wins most real problems.

16 SESSIONS
47 – 62

047 **Linear algebra for ML**
Vectors and matrices · dot products · matrix multiplication · norms · geometric intuition

048 **Calculus and optimization**
Derivatives and partial derivatives · gradients · the chain rule · gradient descent by hand

049 **Probability and statistics**
Distributions · expectation and variance · Bayes · sampling · correlation vs causation

050 **What machine learning is**
Supervised, unsupervised, reinforcement · features and labels · the end-to-end workflow

051 **Train, validate, test**
Splits and why · generalisation · overfitting seen not described · bias–variance · learning curves

052 **scikit-learn**
The estimator API · fit, predict, transform · Pipelines · ColumnTransformer · saving models

053 **Regression**
Linear and polynomial · ridge and lasso · assumptions · residual analysis · interpreting coefficients

054 **Classification**
Logistic regression · decision boundaries · multiclass strategies · probability calibration

055 **Decision trees**
Splitting criteria · depth and pruning · overfitting · interpretability and feature importance

056 **Ensembles**
Bagging · random forests · out-of-bag error · voting and stacking · variance reduction

057 **Gradient boosting**
XGBoost · LightGBM · CatBoost · key hyperparameters · why boosting still wins on tabular data

058 **Evaluation**
Confusion matrix · precision, recall, F1 · ROC-AUC vs PR-AUC · choosing the metric for the business

059 **Validation and tuning**
k-fold and stratification · grid vs random vs Bayesian search · Optuna · nested CV

060 **Data leakage and imbalance**

Leakage patterns · temporal splits · group splits · SMOTE · class weights · threshold tuning

061 **Feature engineering**

Encoding categoricals · scaling · binning · interactions · date features · target encoding

062 **Unsupervised learning**

k-means · DBSCAN · hierarchical clustering · PCA · UMAP · anomaly detection

PROJECT 4

End-to-end machine learning

A real problem on real data. No Titanic, no Iris — a dataset with missing values and lies in it.

DELIVERABLE

Reproducible pipeline · written evaluation stating honestly what the model cannot do

PROVES

Maths · scikit-learn · evaluation · feature engineering

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
05 Deep learning

Neural networks, without the magic.

12 SESSIONS
63 – 74

---

063 **A neural network by hand**
Neurons and layers · weights and biases · activation functions · a forward pass in pure NumPy

064 **Backpropagation**
The chain rule applied · gradients through layers · why it works · vanishing and exploding gradients

065 **PyTorch fundamentals**
Tensors · devices and dtypes · autograd · the computational graph · nn.Module

066 **The training loop**
Datasets and batches · epochs · forward, backward, step · evaluation mode · checkpointing

067 **Loss functions**
MSE and MAE · cross-entropy · focal loss · custom losses · matching the loss to the task

068 **Optimizers and schedules**
SGD and momentum · Adam and AdamW · learning-rate finding · warmup · cosine decay

069 **Regularization**
Dropout · weight decay · batch and layer normalization · early stopping · label smoothing

070 **Data pipelines**
Dataset and DataLoader · transforms · augmentation · class balance · workers and prefetching

071 **Hardware and speed**
GPU memory · batch size · mixed precision · gradient accumulation · Colab and Kaggle in practice

072 **Debugging training runs**
Loss will not move · NaNs · overfitting a single batch as a sanity check · gradient checks

073 **Experiment discipline**
Seeds and determinism · baselines first · ablations · reading a training curve honestly

074 **train a network end to end** LAB
Full build: dataset, model, training, evaluation, write-up

PROJECT 5

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
06 Computer Vision

Detection, video, and the edge.

16 SESSIONS

75 – 90

075 **Images as data**

Pixels and channels · color spaces · resizing and interpolation · normalization · OpenCV basics

076 **Classical computer vision**

Thresholding · filters and edges · contours · morphology · when classical beats deep learning

077 **Convolutions**

Kernels · stride and padding · receptive field · pooling · what filters actually learn

078 **CNN architectures**

LeNet to ResNet · skip connections · EfficientNet · Vision Transformers in brief

079 **Transfer learning**

Pretrained backbones · freezing and unfreezing · fine-tuning · what to do when data is scarce

080 **image classification** LAB

End-to-end build on a real image set, with augmentation and honest evaluation

081 **Data collection and labeling**

Sourcing images · class balance · CVAT, Roboflow, Label Studio · label quality and agreement

082 **Object detection concepts**

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

083 **YOLO in practice**

Architecture overview · Ultralytics · inference · confidence and NMS thresholds

084 **Training a custom detector**

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

085 **Segmentation**

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

086 **Pose and keypoints**

Human pose estimation · landmark detection · skeletons · practical applications

087 **Video and tracking**

Frames vs streams · optical flow · ByteTrack and DeepSORT · re-identification across cameras

088 **Live camera systems**

RTSP and ONVIF · buffering and dropped frames · latency budgets · threading and queues

089 **Edge deployment**

ONNX · TensorRT · quantization and pruning · Jetson and CPU inference · benchmarking FPS

090 **CV build clinic** LAB

Supervised build on the student's own dataset and detector

PROJECT 6

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

OpenCV · CNNs · 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
07 NLP and Transformers

Text, embeddings, fine-tuning.

12 SESSIONS
91 – 102

091 **Text as data**
Normalization · tokenization · BPE and WordPiece · vocabularies · Unicode, and what breaks with Urdu

092 **Representations**
Bag of words · TF-IDF · word2vec and GloVe · sentence transformers · similarity

093 **Classical NLP tasks**
Text classification · named entity recognition · sentiment · topic modeling · spaCy

094 **Sequence models**
RNNs and LSTMs · why they struggled · seq2seq · the problem attention was invented to solve

095 **Attention**
Queries, keys and values · self-attention · multi-head attention · positional encoding

096 **The transformer**
Encoder and decoder · BERT vs GPT families · pretraining objectives · scaling

097 **Hugging Face**
The model hub · tokenizers · pipelines · datasets · reading a model card properly

098 **Fine-tuning**
Task heads · training arguments · freezing layers · catastrophic forgetting · overfitting small data

099 **Parameter-efficient tuning**
LoRA · QLoRA · adapters · when full fine-tuning is unnecessary · merging weights

100 **Evaluating NLP systems**
Accuracy and F1 · BLEU and ROUGE · perplexity · human evaluation · designing a test set

101 **Multilingual and Urdu**
Script and tokenization problems · low-resource strategies · transliteration · code-switching

102 **NLP build clinic** LAB
Supervised build and evaluation of the student's own fine-tuned model

PROJECT 7

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

Tokenization · 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

08 Agentic AI and LLMs

RAG, tools, agents, guardrails.

16 SESSIONS

103 – 118

103 **How LLMs work**

Pretraining and next-token prediction · instruction tuning · RLHF and DPO · what emerges and what does not

104 **Choosing a model**

Hosted vs open weights · size vs cost vs latency · licensing · reading benchmarks sceptically

105 **Prompting as engineering**

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

106 **Structured output**

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

107 **Tokens, context and cost**

Context windows · truncation strategies · streaming · latency · cost per request modelled properly

108 **Embeddings and vector stores**

Embedding models · similarity metrics · pgvector, Qdrant, Chroma · indexing and recall

109 **RAG I – ingestion**

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

110 **RAG II – retrieval**

Dense and hybrid retrieval · BM25 · reranking · query rewriting · returning citations

111 **RAG III – failure modes**

Hallucination · stale data · retrieval misses · chunk boundary problems · measuring retrieval quality

112 **Evaluating LLM systems**

Golden sets · LLM-as-judge and its limits · regression testing · human review loops

113 **Tool and function calling**

Tool schemas · execution and error handling · sandboxing · deciding when a tool is needed

114 **Agent architectures**

ReAct · planning · reflection · task decomposition · loop control and stopping conditions

115 **Memory and state**

Short vs long-term memory · conversation history · summarization · user profiles and personalisation

116 **Multi-agent systems**

Orchestration · hand-offs · supervisor patterns · shared state · when it is plainly overkill

117 **Safety and guardrails**

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

118 **agent build clinic** LAB

Supervised build of a retrieval-and-tools application

PROJECT 8

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
09 DevOps and MLOps

Ship it, run it, and know when it breaks.

12 SESSIONS
119 – 130

119 **Why notebooks do not ship**
Reproducibility · hidden state · the research-to-production gap · what production actually demands

120 **Docker I**
Images and containers · writing a Dockerfile · layers and caching · volumes · docker compose

121 **Docker II**
Multi-stage builds · slim images · GPU containers · registries · image size and security

122 **Linux for servers**
systemd services · cron · nginx as a reverse proxy · log files · basic server monitoring

123 **CI/CD**
GitHub Actions · test, lint and build pipelines · environments and secrets · automated deployment

124 **Cloud fundamentals**
Compute, object storage and networking · IAM and least privilege · regions · controlling cost

125 **Infrastructure as code**
Why click-ops fails · Terraform basics · environments · secrets management · Vault and cloud equivalents

126 **Experiment tracking**
MLflow · Weights & Biases · runs, params, metrics, artifacts · comparing and sharing experiments

127 **Model and data versioning**
Model registry · staging and promotion · DVC · data lineage · rolling back a bad model

128 **Serving models**
FastAPI plus a model · batch vs real-time · ONNX Runtime · concurrency · load testing

129 **Monitoring in production**
Metrics, logs and traces · data drift · concept drift · alerting · Prometheus and Grafana

130 **Retraining and orchestration**
Scheduled retraining · feature stores · Airflow and Prefect · pipeline failure handling

PROJECT 9

A deployed, monitored service

Take an earlier model and put it into production properly, with everything a real team expects.

DELIVERABLE

Dockerized · CI/CD pipeline · deployed to cloud · tracked experiments · monitoring and alerts

PROVES

Docker · CI/CD · cloud · MLflow · model registry · monitoring

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

MODULE 10

LLMOps and capstone

Running AI systems that do not behave deterministically.

10 SESSIONS

131 – 140

131 **LLMOps – what changes**

Non-determinism · no fixed test set · cost as a first-class metric · why MLOps habits do not fully transfer

132 **Observability for LLMs**

Tracing with LangSmith or Langfuse · logging prompts and completions · prompt version control

133 **Evaluation in production**

Online evaluation · A/B testing prompts and models · user feedback loops · drift in prompt performance

134 **Cost, caching and reliability**

Semantic caching · model routing · fallbacks and degradation · handling rate limits and outages

135 **Serving open models**

vLLM · quantization for inference · GPU sizing · the self-host vs API decision, costed

136 **Security and compliance**

Data residency · PII redaction · audit logs · tenant isolation · what enterprise buyers will ask

137 **capstone build** LAB

Supervised build time with instructor review

138 **capstone build** LAB

Supervised build time with instructor review

139 **Career**

CV and GitHub polish · portfolio narrative · explaining your own projects · technical interview practice

140 **Demo day** DEMO

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

CAPSTONE

Deployed, documented, defended

A system of the student's choosing that solves a real problem for a real user.

DELIVERABLE

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

PROVES

Everything above

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.