Admissions & Support: support@mockattempt.com
Phone / WhatsApp: +91 95400 07917
Course Code: DSAI • **Updated:** 05 Nov 2026
https://mockattempt.com/courses/applied-data-science-genai-mlops-engineering

DSAI • DATA SCIENCE & AI | **BEGINNER TO ADVANCED** | **100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER**

Applied Data Science, GenAI & MLOps Engineering

Master Python data workflows, machine learning modeling, LLM RAG agents, and FastAPI model serving with verified topic assessments.

Duration: 18–22 Weeks • Live + Hands-On Projects (211 Hrs) • **Modules:** 14 • **Live Topics:** 42 • **Subtopics:** 336+ • **Labs & Tasks:** 84 • **Mocks / AI Vivas:** 84 / 84 • **Projects:** 14 Module + 1 Live Client Prod

All-Inclusive Direct Fee (Incl. 18% GST • Zero Hidden Charges): Total Payable: ₹6,999 (Net Tuition ₹5,931 + Included 18% GST ₹1,068) • **Split Option:** ₹1,999 Lock + 2 × ₹2,500 • **Effective ROI:** ~₹33/Guided Hr (~₹167/Live Topic)

Faculty: 100% Active Working IT Professionals • **Live Batch Tracks (7:00 AM – 8:00 PM IST Flexible Slots):** Weekday Live Online Batch (Mon – Fri • 1 Hr/Day • 7:00 AM – 8:00 PM) (Monday to Friday (1 Hour Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 1-Hour Slot Based on Trainer Availability) • 18–20 Weeks Fast-Track) | Weekend Live Online Batch (Sat – Sun • 2 Hrs/Day • 7:00 AM – 8:00 PM) (Saturday & Sunday (2 Hours Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 2-Hour Slot Based on Trainer Availability) • 20–22 Weeks Weekend Track)

✓ **Trainer LeetCode & HackerRank Mentorship (Python 3.12 & SQL): Live trainer support to crack LeetCode & HackerRank problems, optimize complexity, and handle edge cases.**

Core Technologies & Engineering Stack: Python • PyTorch • Scikit-Learn • FastAPI • LangChain • PostgreSQL pgvector

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Train, evaluate, and serve ML models behind FastAPI microservices
2. Build grounded RAG pipelines with hybrid vector search & evaluation metrics
3. Reproducible Python data pipelines (`NumPy`, `Pandas`, feature stores) with strict train/validation leakage prevention.
4. Classical ML & Deep Learning (`Scikit-Learn`, `XGBoost`, `PyTorch`) tuned via cross-validation, ROC-AUC, and SHAP explainability.
5. Production Generative AI: chunking strategies, embeddings, `pgvector` hybrid retrieval, RAG grounding, and hallucination evals.
6. Crack LeetCode and HackerRank problems in Python 3.12 & SQL with live trainer mentorship on algorithmic/query patterns, edge cases, and optimal complexity.
7. Low-latency model & LLM serving via async FastAPI endpoints with token streaming, guardrails, and cost/latency telemetry.
8. Vectorized NumPy, Pandas Pipelines & Statistical Validation — Clean messy million-row datasets 50× faster and separate real business signals from random noise.

Eligibility & Prerequisites: No prior Applied Data Science, GenAI & MLOps Engineering industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews. • Dedication to write original code in your GitHub portfolio and complete the mentor-guided Live Client Production Sprint.

2. DETAILED ZERO-TO-EXPERT MODULE, TOPIC & SUBTOPIC SYLLABUS (14 MODULES • 42 LIVE TOPICS • 336+ SUBTOPICS)

Progressive Zero-to-Expert Mastery Stages: Modules 1–3: Stage 1: Day-1 Foundations & Core Mental Models (9 Topics / 72 Subtopics) • Modules 4–7: Stage 2: Applied Engineering, Data & API Workflows (12 Topics / 96 Subtopics) • Modules 8–11: Stage 3: Internals, Concurrency, Security & Performance (12 Topics / 96 Subtopics) • Modules 12–14: Stage 4: Production Architecture, Testing, CI/CD & Expert Mastery (9 Topics / 72 Subtopics)

MODULE 1 OF 14 • STAGE 1 • 25 LIVE + LAB HOURS • 1 TOPICS (8 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Python Data Structures, Vectorization & Feature Engineering

NumPy memory layouts, Pandas pipelines, and statistical validation.

Objectives: ✓ NumPy Broadcasting & High-Performance Data Pipelines — Eliminating slow Python loops with contiguous C-array vectorization. • ✓ Build, verify, and ship Module Capstone: Real-Time Credit Risk Feature Pipeline & FastAPI Scorer.

1.1 NumPy Broadcasting & High-Performance Data Pipelines [45 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Vectorized Computation • NumPy executes element • Security Hygiene & Data Safety Rules for NumPy Broadcasting & High • Performance Optimization, Complexity & Latency Impact of NumPy Broadcasting & High • Runnable Code Implementation (PYTHON) • Measure wall • Verify zero • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG

Purpose: To understand and apply NumPy Broadcasting & High-Performance Data Pipelines in real Applied Data Science, GenAI & MLOps Engineering systems — specifically: Eliminating s...

What Topic Is Next → Module 1 Capstone (Real-Time Credit Risk Feature Pipeline & FastAPI Scorer) → Next Topic 2.1: Pandas DataFrame Indexing, GroupBy Aggregations & Memory Optimization

Module 1 Hands-On Capstone Project (100 Marks): Real-Time Credit Risk Feature Pipeline & FastAPI Scorer — Engineer a vectorized feature pipeline and deploy a sub-25ms FastAPI credit-risk scoring endpoint with drift metrics.

MODULE 2 OF 14 • STAGE 1 • 16 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Data Wrangling with Pandas, Polars & Analytical Window SQL

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Data Wrangling with Pandas, Polars & Analytical Window SQL with hands-on labs and capstone build.

Objectives: ✓ Pandas DataFrame Indexing, GroupBy Aggregations & Memory Optimization — Purpose: Clean messy multi-million-row CSV/Parquet datasets fast without exhausting RAM. • ✓ Analytical SQL for Data Scientists: CTEs, Window Functions & Funnel Queries — Purpose: Extract ML-ready customer feature tables directly inside PostgreSQL/DuckDB. • ✓ High-Speed Lazy Dataframes with Polars & DuckDB Local Lakehouse — Purpose: Process 20GB+ datasets on a standard laptop using Rust/C++ vectorized query engines. • ✓ Automated Ingestion Pipelines, Pydantic Data Schemas & Idempotent Backfills — Purpose: Ingest partner APIs and webhook JSON logs cleanly into partitioned feature tables.

2.1 Pandas DataFrame Indexing, GroupBy Aggregations & Memory Optimization [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Vectorized Filtering, .loc/.iloc & Method Chaining (.assign, .pipe) • Downcasting Numeric Dtypes, Categoricals & Parquet Columnar I/O • Handling Missing Values, Duplicates & Timezone-Aware Timestamps • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Pandas DataFrame Indexing

Purpose: To understand and apply Pandas DataFrame Indexing, GroupBy Aggregations & Memory Optimization in real Applied Data Science, GenAI & MLOps Engineering systems — specifical...

What Topic Is Next → Next Topic 2.2: Analytical SQL for Data Scientists: CTEs, Window Functions & Funnel Queries

2.2 Analytical SQL for Data Scientists: CTEs, Window Functions & Funnel Queries [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: ROW_NUMBER, LAG, LEAD & Rolling Moving-Average Windows • Modular CTE Pipelines for Cohort Retention & Conversion Funnels • Point-in-Time Joins to Prevent Future Data Leakage • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Analytical SQL for Data Scientists

Purpose: To understand and apply Analytical SQL for Data Scientists: CTEs, Window Functions & Funnel Queries in real Applied Data Science, GenAI & MLOps Engineering systems — spec...

What Topic Is Next → Next Topic 2.3: High-Speed Lazy Dataframes with Polars & DuckDB Local Lakehouse

2.3 High-Speed Lazy Dataframes with Polars & DuckDB Local Lakehouse [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Polars LazyFrame Query Optimization (Predicate & Projection Pushdown) • Zero-Copy Apache Arrow Interop Between DuckDB, Polars & PyTorch • Data Quality Contracts (Schema, Null Thresholds & Range Assertions) • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for High-Speed Lazy Dataframes with Polars

Purpose: To understand and apply High-Speed Lazy Dataframes with Polars & DuckDB Local Lakehouse in real Applied Data Science, GenAI & MLOps Engineering systems — specifically: Pr...

What Topic Is Next → Next Topic 2.4: Automated Ingestion Pipelines, Pydantic Data Schemas & Idempotent Backfills

2.4 Automated Ingestion Pipelines, Pydantic Data Schemas & Idempotent Backfills [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Pydantic v2 Schema Validation on Dirty Partner Payloads • Date-Partitioned Idempotent Backfills (overwrite partition) • Feature Store Concepts (Offline Training Parity vs Online Serving) • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG

• Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Automated Ingestion Pipelines

Purpose: To understand and apply Automated Ingestion Pipelines, Pydantic Data Schemas & Idempotent Backfills in real Applied Data Science, GenAI & MLOps Engineering systems — spec...

What Topic Is Next → Module 2 Capstone (Automated Multi-Source E-Commerce Cohort & Retention Data Pipeline) → Next Topic 3.1: Diagnostic EDA, Outlier Detection & Feature Correlation Forensics

Module 2 Hands-On Capstone Project (100 Marks): Automated Multi-Source E-Commerce Cohort & Retention Data Pipeline — Design, implement, test, and document 'Automated Multi-Source E-Commerce Cohort & Retention Data Pipeline' applying every production concept from Module 2 (Data Wrangling with Pandas, Polars...

MODULE 3 OF 14 • STAGE 1 • 16 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Exploratory Data Analysis, Probability Distributions & A/B Hypothesis Testing

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Exploratory Data Analysis, Probability Distributions & A/B Hypothesis Testing with hands-on labs and capstone build.

Objectives: ✓ Diagnostic EDA, Outlier Detection & Feature Correlation Forensics — Purpose: Uncover data skew, Simpson's paradox, and collinearity before fitting a single model. • ✓ Probability Distributions, Bayes' Theorem & Confidence Intervals — Purpose: Quantify uncertainty around churn rates, fraud probabilities, and demand forecasts. • ✓ Statistical A/B Testing: Sample Size Power, p-Values & Multiple-Testing Guardrails — Purpose: Design valid online product experiments that distinguish real lift from random noise. • ✓ Causal Uplift Thinking, Guardrail Metrics & Network Interference Checks — Purpose: Ensure product launches improve net retention without hurting latency or support tickets.

3.1 Diagnostic EDA, Outlier Detection & Feature Correlation Forensics

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Quantile/IQR vs Robust MAD Outlier Diagnostics • Pearson vs Spearman Rank Correlation & Variance Inflation (VIF) • Executive Visual Storytelling (Distributions, Calibration & Cohort Heatmaps) • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Diagnostic EDA

Purpose: To understand and apply Diagnostic EDA, Outlier Detection & Feature Correlation Forensics in real Applied Data Science, GenAI & MLOps Engineering systems — specifically:...

What Topic Is Next → Next Topic 3.2: Probability Distributions, Bayes' Theorem & Confidence Intervals

3.2 Probability Distributions, Bayes' Theorem & Confidence Intervals

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Bernoulli, Binomial, Poisson, Normal & Log-Normal Business Processes • Bayesian Updating (Prior -> Likelihood -> Posterior) • Bootstrap Resampling Confidence Intervals • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Probability Distributions

Purpose: To understand and apply Probability Distributions, Bayes' Theorem & Confidence Intervals in real Applied Data Science, GenAI & MLOps Engineering systems — specifically: Q...

What Topic Is Next → Next Topic 3.3: Statistical A/B Testing: Sample Size Power, p-Values & Multiple-Testing Guardrails

3.3 Statistical A/B Testing: Sample Size Power, p-Values & Multiple-Testing Guardrails

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Statistical Power (1 - Beta), Minimum Detectable Effect & Sample Sizing • Welch's t-Test, Mann-Whitney U & Chi-Square Proportion Tests • CUPED Variance Reduction & Benjamini-Hochberg FDR Correction • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Statistical A/B Testing

Purpose: To understand and apply Statistical A/B Testing: Sample Size Power, p-Values & Multiple-Testing Guardrails in real Applied Data Science, GenAI & MLOps Engineering systems...

What Topic Is Next → Next Topic 3.4: Causal Uplift Thinking, Guardrail Metrics & Network Interference Checks

3.4 Causal Uplift Thinking, Guardrail Metrics & Network Interference Checks

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Primary North-Star Metrics vs Invariant Guardrail Metrics (Latency, Refund Rate) • Sample Ratio Mismatch (SRM) Chi-Square Sanity Check • Observational Selection Bias vs Randomized Controlled Trials • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Causal Uplift Thinking

Purpose: To understand and apply Causal Uplift Thinking, Guardrail Metrics & Network Interference Checks in real Applied Data Science, GenAI & MLOps Engineering systems — specific...

What Topic Is Next → Module 3 Capstone (Product Experimentation & Statistical A/B Test Decision Engine) → Next Topic 4.1: ColumnTransformer Pipelines, Target Encoding & Leakage Prevention

Module 3 Hands-On Capstone Project (100 Marks): Product Experimentation & Statistical A/B Test Decision Engine — Design, implement, test, and document 'Product Experimentation & Statistical A/B Test Decision Engine' applying every production concept from Module 3 (Exploratory Data Analysis, Probability...

MODULE 4 OF 14 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Leak-Free Feature Engineering, Scikit-Learn Pipelines & Imbalanced Classification

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Leak-Free Feature Engineering, Scikit-Learn Pipelines & Imbalanced Classification with hands-on labs and capstone build.

Objectives: ✓ ColumnTransformer Pipelines, Target Encoding & Leakage Prevention — Package preprocessing and modeling into a single serializable Scikit-Learn Pipeline. • ✓ Regularized Linear Models (Ridge, Lasso, ElasticNet) & Cross-Validation Strategies — Build fast, interpretable baseline models and choose time/group-aware CV splits. • ✓ Imbalanced Learning: PR-AUC, Cost-Sensitive Thresholds & Probability Calibration — Optimize fraud, medical, and churn classifiers where positive cases are < 2% of rows. • ✓ Build, verify, and ship Module Capstone: End-to-End Credit Default & Churn Prediction Pipeline.

4.1 ColumnTransformer Pipelines, Target Encoding & Leakage Prevention [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Fitting Scalers/Imputers Strictly Inside Cross-Validation Folds • Out-of-Fold Regularized Target Encoding for High-Cardinality Categoricals • Cyclical Time Features (sin/cos Hour/Day) & Lag Aggregations • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for ColumnTransformer Pipelines

Purpose: To understand and apply ColumnTransformer Pipelines, Target Encoding & Leakage Prevention in real Applied Data Science, GenAI & MLOps Engineering systems — specifically:...

What Topic Is Next → Next Topic 4.2: Regularized Linear Models (Ridge, Lasso, ElasticNet) & Cross-Validation Strategies

4.2 Regularized Linear Models (Ridge, Lasso, ElasticNet) & Cross-Validation Strategies [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: L1 Sparsity vs L2 Shrinkage & Feature Scaling Necessity • StratifiedKFold, GroupKFold & TimeSeriesSplit • Bias–Variance Diagnostics via Learning & Validation Curves • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Regularized Linear Models (Ridge, Lasso, ElasticNet)

Purpose: To understand and apply Regularized Linear Models (Ridge, Lasso, ElasticNet) & Cross-Validation Strategies in real Applied Data Science, GenAI & MLOps Engineering systems...

What Topic Is Next → Next Topic 4.3: Imbalanced Learning: PR-AUC, Cost-Sensitive Thresholds & Probability Calibration

4.3 Imbalanced Learning: PR-AUC, Cost-Sensitive Thresholds & Probability Calibration [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Why Accuracy & ROC-AUC Mislead on 99:1 Imbalanced Datasets • Class Weights & Cost-Matrix Decision Threshold Optimization • Platt Scaling & Isotonic Probability Calibration (Brier Score) • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Imbalanced Learning

Purpose: To understand and apply Imbalanced Learning: PR-AUC, Cost-Sensitive Thresholds & Probability Calibration in real Applied Data Science, GenAI & MLOps Engineering systems —...

What Topic Is Next → Module 4 Capstone (End-to-End Credit Default & Churn Prediction Pipeline) → Next Topic 5.1: Gradient Boosting Math, Histogram Split Finding & XGBoost/LightGBM Tuning

Module 4 Hands-On Capstone Project (100 Marks): End-to-End Credit Default & Churn Prediction Pipeline — Design, implement, test, and document 'End-to-End Credit Default & Churn Prediction Pipeline' applying every production concept from Module 4 (Leak-Free Feature Engineering, Scikit-Learn Pip...

MODULE 5 OF 14 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Gradient Boosted Ensembles (XGBoost, LightGBM), Optuna & SHAP Explainability

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Gradient Boosted Ensembles (XGBoost, LightGBM), Optuna & SHAP Explainability with hands-on labs and capstone build.

Objectives: ✓ Gradient Boosting Math, Histogram Split Finding & XGBoost/LightGBM Tuning — Master the #1 tabular ML algorithm family used across FinTech, retail, and ad-tech. • ✓ Bayesian Hyperparameter Optimization with Optuna & Early Stopping — Search hyperparameter spaces intelligently in a fraction of GridSearch compute cost. • ✓ Model Explainability with TreeSHAP, Permutation Importance & Fairness Audits — Generate regulator-ready per-customer prediction explanations and detect subgroup bias. • ✓ Build, verify, and ship Module Capstone: Explainable Real-Time Fraud Scoring Engine with SHAP Reason Codes.

5.1 Gradient Boosting Math, Histogram Split Finding & XGBoost/LightGBM Tuning [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Residual Gradient Fitting, Second-Order Hessians & Leaf-Wise vs Depth-Wise Trees • Regularization Controls (learning_rate, subsample, colsample_bytree, min_child_weight) • Native Missing-Value Routing & Monotonic Constraints • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery •

Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Gradient Boosting Math

Purpose: To understand and apply Gradient Boosting Math, Histogram Split Finding & XGBoost/LightGBM Tuning in real Applied Data Science, GenAI & MLOps Engineering systems — specif...

What Topic Is Next → Next Topic 5.2: Bayesian Hyperparameter Optimization with Optuna & Early Stopping

5.2 Bayesian Hyperparameter Optimization with Optuna & Early Stopping [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Tree-Structured Parzen Estimator (TPE) & Pruning Unpromising Trials • Early Stopping on Validation Holdout vs Nested CV • Ensembling & Stacking Diverse Model Families • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Bayesian Hyperparameter Optimization with Optuna

Purpose: To understand and apply Bayesian Hyperparameter Optimization with Optuna & Early Stopping in real Applied Data Science, GenAI & MLOps Engineering systems — specifically:...

What Topic Is Next → Next Topic 5.3: Model Explainability with TreeSHAP, Permutation Importance & Fairness Audits

5.3 Model Explainability with TreeSHAP, Permutation Importance & Fairness Audits [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Game-Theoretic SHAP Values (Global Summary vs Local Waterfall Plots) • Permutation Feature Importance vs Biased Impurity Split Counts • Subgroup Slice Evaluation & Disparate Impact Audits • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Model Explainability with TreeSHAP

Purpose: To understand and apply Model Explainability with TreeSHAP, Permutation Importance & Fairness Audits in real Applied Data Science, GenAI & MLOps Engineering systems — spe...

What Topic Is Next → Module 5 Capstone (Explainable Real-Time Fraud Scoring Engine with SHAP Reason Codes) → Next Topic 6.1: K-Means, HDBSCAN Density Clustering & Behavioral RFM Segmentation

Module 5 Hands-On Capstone Project (100 Marks): Explainable Real-Time Fraud Scoring Engine with SHAP Reason Codes — Design, implement, test, and document 'Explainable Real-Time Fraud Scoring Engine with SHAP Reason Codes' applying every production concept from Module 5 (Gradient Boosted Ensembles (XGBoost...

MODULE 6 OF 14 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Unsupervised Learning, Dimensionality Reduction, Anomaly Detection & Forecasting

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Unsupervised Learning, Dimensionality Reduction, Anomaly Detection & Forecasting with hands-on labs and capstone build.

Objectives: ✓ K-Means, HDBSCAN Density Clustering & Behavioral RFM Segmentation — Group millions of unlabeled users or SKUs into actionable business segments. • ✓ PCA, UMAP Visualization & Unsupervised Anomaly Detection (Isolation Forest) — Compress wide sensor/embedding spaces and catch novel zero-day anomalies without labels. • ✓ Time-Series Decomposition, Walk-Forward Backtesting & Multi-Step Forecasting — Forecast daily inventory demand, cloud spend, and revenue across multiple horizons. • ✓ Build, verify, and ship Module Capstone: RFM Customer Segmentation & Multi-Horizon Demand Forecaster.

6.1 K-Means, HDBSCAN Density Clustering & Behavioral RFM Segmentation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: K-Means++ Initialization, Silhouette Scores & Feature Scaling Sensitivity • Density-Based HDBSCAN for Non-Spherical Clusters & Noise Rejection • Profiling & Naming Clusters into Business Playbooks • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for K-Means

Purpose: To understand and apply K-Means, HDBSCAN Density Clustering & Behavioral RFM Segmentation in real Applied Data Science, GenAI & MLOps Engineering systems — specifically:...

What Topic Is Next → Next Topic 6.2: PCA, UMAP Visualization & Unsupervised Anomaly Detection (Isolation Forest)

6.2 PCA, UMAP Visualization & Unsupervised Anomaly Detection (Isolation Forest) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Singular Value Decomposition (PCA) & Explained Variance Ratios • UMAP Embedding Projections for Visual Cluster Inspection • Isolation Forest & Robust Mahalanobis Anomaly Scoring • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for UMAP Visualization

Purpose: To understand and apply PCA, UMAP Visualization & Unsupervised Anomaly Detection (Isolation Forest) in real Applied Data Science, GenAI & MLOps Engineering systems — spec...

What Topic Is Next → Next Topic 6.3: Time-Series Decomposition, Walk-Forward Backtesting & Multi-Step Forecasting

6.3 Time-Series Decomposition, Walk-Forward Backtesting & Multi-Step Forecasting [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Trend, Multiple Seasonality & Holiday Regressors • Expanding-Window Walk-Forward Backtesting (MAPE, sMAPE, WAPE) • Global Gradient-Boosted Lag Forecasters Across 10,000+ SKUs • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Time-Series Decomposition

Purpose: To understand and apply Time-Series Decomposition, Walk-Forward Backtesting & Multi-Step Forecasting in real Applied Data Science, GenAI & MLOps Engineering systems — spe...

What Topic Is Next → Module 6 Capstone (RFM Customer Segmentation & Multi-Horizon Demand Forecaster) → Next Topic 7.1: PyTorch Tensors, Dynamic Autograd Graphs & Custom Dataset/DataLoader

Module 6 Hands-On Capstone Project (100 Marks): RFM Customer Segmentation & Multi-Horizon Demand Forecaster — Design, implement, test, and document 'RFM Customer Segmentation & Multi-Horizon Demand Forecaster' applying every production concept from Module 6 (Unsupervised Learning, Dimensionality Red...

MODULE 7 OF 14 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Deep Learning with PyTorch: Autograd, Neural Architectures & Fine-Tuning

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Deep Learning with PyTorch: Autograd, Neural Architectures & Fine-Tuning with hands-on labs and capstone build.

Objectives: ✓ PyTorch Tensors, Dynamic Autograd Graphs & Custom Dataset/DataLoader — Build GPU-accelerated neural training loops with clean data batching. • ✓ Optimizers (AdamW), Cosine LR Schedules, Mixed Precision (AMP) & Checkpointing — Train neural networks 2x faster in BF16/FP16 with reproducible model checkpoints. • ✓ Transfer Learning, Vision Backbones & ONNX Runtime Quantized Export — Fine-tune pretrained backbones on small client datasets and serve them on low-cost CPUs. • ✓ Build, verify, and ship Module Capstone: PyTorch Multi-Modal Defect & Document Classifier with Mixed Precision.

7.1 PyTorch Tensors, Dynamic Autograd Graphs & Custom Dataset/DataLoader [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Computation Graphs, .backward(), optimizer.zero_grad() & torch.no_grad() • Custom Dataset, Pinned-Memory DataLoaders & Multi-Worker Prefetching • Weight Initialization, BatchNorm/LayerNorm & Dropout Regularization • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for PyTorch Tensors

Purpose: To understand and apply PyTorch Tensors, Dynamic Autograd Graphs & Custom Dataset/DataLoader in real Applied Data Science, GenAI & MLOps Engineering systems — specificall...

What Topic Is Next → Next Topic 7.2: Optimizers (AdamW), Cosine LR Schedules, Mixed Precision (AMP) & Checkpointing

7.2 Optimizers (AdamW), Cosine LR Schedules, Mixed Precision (AMP) & Checkpointing [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: SGD+Momentum vs AdamW Decoupled Weight Decay & Gradient Clipping • Warmup + Cosine Annealing Schedules & Automatic Mixed Precision • Best-Metric Model Checkpointing & Deterministic Seed Reproducibility • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Optimizers (AdamW)

Purpose: To understand and apply Optimizers (AdamW), Cosine LR Schedules, Mixed Precision (AMP) & Checkpointing in real Applied Data Science, GenAI & MLOps Engineering systems — s...

What Topic Is Next → Next Topic 7.3: Transfer Learning, Vision Backbones & ONNX Runtime Quantized Export

7.3 Transfer Learning, Vision Backbones & ONNX Runtime Quantized Export [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Freezing Pretrained Backbones vs Differential Layer Learning Rates • Data Augmentation & Test-Time Calibration • Exporting PyTorch Models to ONNX & INT8 Dynamic Quantization • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Transfer Learning

Purpose: To understand and apply Transfer Learning, Vision Backbones & ONNX Runtime Quantized Export in real Applied Data Science, GenAI & MLOps Engineering systems — specifically...

What Topic Is Next → Module 7 Capstone (PyTorch Multi-Modal Defect & Document Classifier with Mixed Precision) → Next Topic 8.1: Self-Attention Mechanics, Subword Tokenization & Encoder vs Decoder LLMs

Module 7 Hands-On Capstone Project (100 Marks): PyTorch Multi-Modal Defect & Document Classifier with Mixed Precision — Design, implement, test, and document 'PyTorch Multi-Modal Defect & Document Classifier with Mixed Precision' applying every production concept from Module 7 (Deep Learning with PyTorch: Aut...

MODULE 8 OF 14 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Transformers, Tokenization, Dense Embeddings & Parameter-Efficient Fine-Tuning (LoRA)

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Transformers, Tokenization, Dense Embeddings & Parameter-Efficient Fine-Tuning (LoRA) with hands-on labs and capstone build.

Objectives: ✓ Self-Attention Mechanics, Subword Tokenization & Encoder vs Decoder LLMs — Understand exactly how modern Transformers process context windows, tokens, and KV caches. • ✓ Dense Bi-Encoder Embeddings, Cosine Similarity & Cross-Encoder Reranking — Map sentences, resumes, and products into semantic vector spaces. • ✓ Prompt Engineering Systems, Structured JSON Outputs & LoRA / QLoRA Fine-Tuning — Control LLM outputs deterministically and know when prompting beats fine-tuning. • ✓ Build, verify, and ship Module Capstone: Domain-Adapted Support Ticket Triage & Semantic Similarity Model.

8.1 Self-Attention Mechanics, Subword Tokenization & Encoder vs Decoder LLMs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Scaled Dot-Product Multi-Head Attention & Positional Encodings (RoPE) • BPE / WordPiece Tokenizers, Token Costs & Multilingual Fertility • Encoder (BERT/BGE Embeddings) vs Decoder (GPT/Claude/Llama) Selection • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Self-Attention Mechanics

Purpose: To understand and apply Self-Attention Mechanics, Subword Tokenization & Encoder vs Decoder LLMs in real Applied Data Science, GenAI & MLOps Engineering systems — specifi...

What Topic Is Next → Next Topic 8.2: Dense Bi-Encoder Embeddings, Cosine Similarity & Cross-Encoder Reranking

8.2 Dense Bi-Encoder Embeddings, Cosine Similarity & Cross-Encoder Reranking [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Bi-Encoder Embedding Normalization & Dot-Product / Cosine Math • Two-Stage Retrieval: Fast Bi-Encoder Top-50 -> Cross-Encoder Top-5 Rerank • Hard-Negative Mining & Contrastive Embedding Fine-Tuning • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Dense Bi-Encoder Embeddings

Purpose: To understand and apply Dense Bi-Encoder Embeddings, Cosine Similarity & Cross-Encoder Reranking in real Applied Data Science, GenAI & MLOps Engineering systems — specifi...

What Topic Is Next → Next Topic 8.3: Prompt Engineering Systems, Structured JSON Outputs & LoRA / QLoRA Fine-Tuning

8.3 Prompt Engineering Systems, Structured JSON Outputs & LoRA / QLoRA Fine-Tuning [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: System Instructions, Few-Shot Exemplars & Schema-Constrained JSON Mode • When to Use Prompting vs RAG vs LoRA Fine-Tuning • 4-Bit QLoRA Low-Rank Adapter Training & Adapter Merging • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Prompt Engineering Systems

Purpose: To understand and apply Prompt Engineering Systems, Structured JSON Outputs & LoRA / QLoRA Fine-Tuning in real Applied Data Science, GenAI & MLOps Engineering systems — s...

What Topic Is Next → Module 8 Capstone (Domain-Adapted Support Ticket Triage & Semantic Similarity Model) → Next Topic 9.1: Document Parsing, Structure-Aware Chunking & Metadata Enrichment

Module 8 Hands-On Capstone Project (100 Marks): Domain-Adapted Support Ticket Triage & Semantic Similarity Model — Design, implement, test, and document 'Domain-Adapted Support Ticket Triage & Semantic Similarity Model' applying every production concept from Module 8 (Transformers, Tokenization, Dense Em...

MODULE 9 OF 14 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Vector Databases (PostgreSQL pgvector), HNSW Indexing & Hybrid Search

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Vector Databases (PostgreSQL pgvector), HNSW Indexing & Hybrid Search with hands-on labs and capstone build.

Objectives: ✓ Document Parsing, Structure-Aware Chunking & Metadata Enrichment — Prepare PDFs, policies, and codebases into high-signal chunks that retrieve cleanly. • ✓ PostgreSQL pgvector, HNSW vs IVFFlat Indexes & Pre-Filtered Tenant Search — Run vector search right inside ACID PostgreSQL alongside relational RBAC permissions. • ✓ Hybrid Search: Combining Lexical Full-Text BM25 with Reciprocal Rank Fusion (RRF) — Handle exact SKU codes, error IDs, and semantic natural-language queries simultaneously. • ✓ Build, verify, and ship Module Capstone: Multi-Tenant Hybrid Semantic + BM25 Search Engine on pgvector.

9.1 Document Parsing, Structure-Aware Chunking & Metadata Enrichment [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Markdown/Heading-Aware Chunking vs Naive Fixed-Character Splits • Contextual Chunk Headers, Overlap Windows & Parent-Document Links • Idempotent Content-Hash Indexing (SHA-256) • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Document Parsing

Purpose: To understand and apply Document Parsing, Structure-Aware Chunking & Metadata Enrichment in real Applied Data Science, GenAI & MLOps Engineering systems — specifically: P...

What Topic Is Next → Next Topic 9.2: PostgreSQL pgvector, HNSW vs IVFFlat Indexes & Pre-Filtered Tenant Search

9.2 PostgreSQL pgvector, HNSW vs IVFFlat Indexes & Pre-Filtered Tenant Search [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: HNSW Graph Parameters (m, ef_construction, ef_search) & Recall-Latency Tradeoffs • Tenant & Role Pre-Filtering with Iterative Index Scans • Halfvec (FP16) & Binary Quantization Memory Reduction • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for PostgreSQL pgvector

Purpose: To understand and apply PostgreSQL pgvector, HNSW vs IVFFlat Indexes & Pre-Filtered Tenant Search in real Applied Data Science, GenAI & MLOps Engineering systems — specif...

What Topic Is Next → Next Topic 9.3: Hybrid Search: Combining Lexical Full-Text BM25 with Reciprocal Rank Fusion (RRF)

9.3 Hybrid Search: Combining Lexical Full-Text BM25 with Reciprocal Rank Fusion (RRF)

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Why Pure Vector Search Misses Exact Error Codes & Policy Numbers • Reciprocal Rank Fusion (RRF) Score Merging • Measuring Retrieval Quality: Recall@K, MRR & NDCG@10 • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Hybrid Search

Purpose: To understand and apply Hybrid Search: Combining Lexical Full-Text BM25 with Reciprocal Rank Fusion (RRF) in real Applied Data Science, GenAI & MLOps Engineering systems...

What Topic Is Next → Module 9 Capstone (Multi-Tenant Hybrid Semantic + BM25 Search Engine on pgvector) → Next Topic 10.1: Query Rewriting, Multi-Query Expansion & Conversational Memory Compression

Module 9 Hands-On Capstone Project (100 Marks): Multi-Tenant Hybrid Semantic + BM25 Search Engine on pgvector — Design, implement, test, and document 'Multi-Tenant Hybrid Semantic + BM25 Search Engine on pgvector' applying every production concept from Module 9 (Vector Databases (PostgreSQL pgvector),...

MODULE 10 OF 14 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Production Enterprise RAG Systems, Grounded Citations & Safety Guardrails

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Production Enterprise RAG Systems, Grounded Citations & Safety Guardrails with hands-on labs and capstone build.

Objectives: ✓ Query Rewriting, Multi-Query Expansion & Conversational Memory Compression — Resolve follow-up pronouns (\"what about its refund fee?\") into standalone retrieval queries. • ✓ Strict Citation Grounding, Abstention Logic & Faithfulness Verification — Ensure enterprise copilots cite exact source paragraphs and abstain when evidence is absent. • ✓ Prompt Injection Defense, PII Redaction & Semantic Response Caching — Protect GenAI endpoints against jailbreaks, data exfiltration, and runaway API bills. • ✓ Build, verify, and ship Module Capstone: Hallucination-Resistant Enterprise Policy & Compliance RAG Copilot.

10.1 Query Rewriting, Multi-Query Expansion & Conversational Memory Compression

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Conversation-Aware Query Contextualization • Metadata Filter Extraction & Query Routing • Context Window Packing & Lost-in-the-Middle Mitigation • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Query Rewriting

Purpose: To understand and apply Query Rewriting, Multi-Query Expansion & Conversational Memory Compression in real Applied Data Science, GenAI & MLOps Engineering systems — speci...

What Topic Is Next → Next Topic 10.2: Strict Citation Grounding, Abstention Logic & Faithfulness Verification

10.2 Strict Citation Grounding, Abstention Logic & Faithfulness Verification

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Inline Source Span Citations ([Doc#14, Sec 3.2]) • Calibrated \"I Don’t Know\" Abstention on Out-of-Scope Queries • NLI / Judge Faithfulness Checks Before Returning High-Stakes Answers • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Strict Citation Grounding

Purpose: To understand and apply Strict Citation Grounding, Abstention Logic & Faithfulness Verification in real Applied Data Science, GenAI & MLOps Engineering systems — specific...

What Topic Is Next → Next Topic 10.3: Prompt Injection Defense, PII Redaction & Semantic Response Caching

10.3 Prompt Injection Defense, PII Redaction & Semantic Response Caching

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Indirect Prompt Injection Defense on Untrusted Documents/Emails • Input/Output PII Masking (Phone, PAN, Email, Secrets) • Exact & Semantic Redis Caching for High-Frequency Questions • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Prompt Injection Defense

Purpose: To understand and apply Prompt Injection Defense, PII Redaction & Semantic Response Caching in real Applied Data Science, GenAI & MLOps Engineering systems — specifically...

What Topic Is Next → Module 10 Capstone (Hallucination-Resistant Enterprise Policy & Compliance RAG Copilot) → Next Topic 11.1: Deterministic Function Calling, Tool Schemas & Human-in-the-Loop Approvals

Module 10 Hands-On Capstone Project (100 Marks): Hallucination-Resistant Enterprise Policy & Compliance RAG Copilot — Design, implement, test, and document 'Hallucination-Resistant Enterprise Policy & Compliance RAG Copilot' applying every production concept from Module 10 (Production Enterprise RAG Systems...

MODULE 11 OF 14 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Agentic AI Workflows, Function/Tool Calling & Automated LLM Evaluation

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Agentic AI Workflows, Function/Tool Calling & Automated LLM Evaluation with hands-on labs and capstone build.

Objectives: ✓ Deterministic Function Calling, Tool Schemas & Human-in-the-Loop Approvals — Connect MockAttempt AI/LLMs safely to internal SQL, CRM, and ticketing APIs. • ✓ Multi-Step Graph Workflows (Router -> Retriever -> Verifier) vs Free-Form Agents — Design predictable state-machine AI workflows that hit 99% production reliability. • ✓ Automated Golden-Dataset Evals (RAGAS / LLM-as-Judge) in GitHub Actions CI — Prevent prompt edits or model upgrades from silently breaking accuracy. • ✓ Build, verify, and ship Module Capstone: Autonomous Customer Operations Triage Agent with Golden-Set CI Evals.

11.1 Deterministic Function Calling, Tool Schemas & Human-in-the-Loop Approvals

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Typed JSON Schema Tool Definitions & Argument Validation • Read-Only Autonomous Tools vs Human-Approved Mutating Actions • Bounded Step Budgets, Timeout Guards & Loop Detection • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Deterministic Function Calling

Purpose: To understand and apply Deterministic Function Calling, Tool Schemas & Human-in-the-Loop Approvals in real Applied Data Science, GenAI & MLOps Engineering systems — speci...

What Topic Is Next → Next Topic 11.2: Multi-Step Graph Workflows (Router -> Retriever -> Verifier) vs Free-Form Agents

11.2 Multi-Step Graph Workflows (Router -> Retriever -> Verifier) vs Free-Form Agents

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Explicit Directed State Graphs Over Unconstrained Autonomous Loops • Structured State Checkpointing & Resumable Workflows • Fallback Model Routing (Primary Frontier Tier -> Fast Backup Tier) • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Multi-Step Graph Workflows (Router -> Retriever -> V...

Purpose: To understand and apply Multi-Step Graph Workflows (Router -> Retriever -> Verifier) vs Free-Form Agents in real Applied Data Science, GenAI & MLOps Engineering systems —...

What Topic Is Next → Next Topic 11.3: Automated Golden-Dataset Evals (RAGAS / LLM-as-Judge) in GitHub Actions CI

11.3 Automated Golden-Dataset Evals (RAGAS / LLM-as-Judge) in GitHub Actions CI

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Curating Golden Evaluation Sets (Happy Paths, Adversarial & Abstentions) • Measuring Context Precision, Context Recall, Faithfulness & Answer Relevance • Rubric-Calibrated LLM-as-Judge with Pairwise Tie-Breaking • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Automated Golden-Dataset Evals (RAGAS / LLM-as-Judge...

Purpose: To understand and apply Automated Golden-Dataset Evals (RAGAS / LLM-as-Judge) in GitHub Actions CI in real Applied Data Science, GenAI & MLOps Engineering systems — speci...

What Topic Is Next → Module 11 Capstone (Autonomous Customer Operations Triage Agent with Golden-Set CI Evals) → Next Topic 12.1: Async FastAPI Model Serving, Lifespan Warmup & SSE Token Streaming

Module 11 Hands-On Capstone Project (100 Marks): Autonomous Customer Operations Triage Agent with Golden-Set CI Evals — Design, implement, test, and document 'Autonomous Customer Operations Triage Agent with Golden-Set CI Evals' applying every production concept from Module 11 (Agentic AI Workflows, Function/...

MODULE 12 OF 14 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Production MLOps: Async FastAPI Serving, SSE Token Streaming & Drift Monitoring

Production-grade Applied Data Science, GenAI & MLOps Engineering engineering module covering Production MLOps: Async FastAPI Serving, SSE Token Streaming & Drift Monitoring with hands-on labs and capstone build.

Objectives: ✓ Async FastAPI Model Serving, Lifespan Warmup & SSE Token Streaming — Expose ML and GenAI pipelines as low-latency, high-concurrency production APIs. • ✓ Experiment Tracking, Model Registry (MLflow) & Reproducible Docker Packaging — Version datasets, hyperparameters, prompts, and container artifacts with full lineage. • ✓ Data/Concept Drift Detection (PSI), Token Cost Governance & Retraining Triggers — Monitor live models after launch so silent distribution shifts never erode business ROI. • ✓ Build, verify, and ship Module Capstone: Containerized ML + GenAI Inference Service with Drift & Cost Telemetry.

12.1 Async FastAPI Model Serving, Lifespan Warmup & SSE Token Streaming [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Loading Model Artifacts Once in FastAPI Lifespan Context • Server-Sent Events (SSE) Token Streaming for Sub-300ms Time-to-First-Token • Dynamic Request Batching & Async Worker Offloading • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Async FastAPI Model Serving

Purpose: To understand and apply Async FastAPI Model Serving, Lifespan Warmup & SSE Token Streaming in real Applied Data Science, GenAI & MLOps Engineering systems — specifically:...

What Topic Is Next → Next Topic 12.2: Experiment Tracking, Model Registry (MLflow) & Reproducible Docker Packaging

12.2 Experiment Tracking, Model Registry (MLflow) & Reproducible Docker Packaging [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Logging Params, Metrics, Feature Schemas & SHAP Artifacts in MLflow • Champion–Challenger Shadow Mode & Canary Traffic Splitting • Slim Multi-Stage Python Dockerfiles with Pinned Lockfiles • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Experiment Tracking

Purpose: To understand and apply Experiment Tracking, Model Registry (MLflow) & Reproducible Docker Packaging in real Applied Data Science, GenAI & MLOps Engineering systems — spe...

What Topic Is Next → Next Topic 12.3: Data/Concept Drift Detection (PSI), Token Cost Governance & Retraining Triggers

12.3 Data/Concept Drift Detection (PSI), Token Cost Governance & Retraining Triggers [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Population Stability Index (PSI) & Kolmogorov-Smirnov Feature Drift Alerts • Delayed Ground-Truth Feedback Loops & Live Calibration Tracking • Per-Tenant Token Quotas, Latency p95 Dashboards & Automated Retraining • Under-the-Hood Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle • Boundary Validation, OAuth2 scopes, prompt-injection guardrails & token-rate quotas & Failure Recovery • Automated pytest-asyncio, httpx AsyncClient & RAG faithfulness eval Verification & Senior Viva Trade-Offs • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Data/Concept Drift Detection (PSI)

Purpose: To understand and apply Data/Concept Drift Detection (PSI), Token Cost Governance & Retraining Triggers in real Applied Data Science, GenAI & MLOps Engineering systems —...

What Topic Is Next → Module 12 Capstone (Containerized ML + GenAI Inference Service with Drift & Cost Telemetry) → Next Topic 13.1: Chunking Strategies, Metadata Filtering & Hybrid Keyword + Vector Retrieval

Module 12 Hands-On Capstone Project (100 Marks): Containerized ML + GenAI Inference Service with Drift & Cost Telemetry — Design, implement, test, and document 'Containerized ML + GenAI Inference Service with Drift & Cost Telemetry' applying every production concept from Module 12 (Production MLOps: Async FastA...

MODULE 13 OF 14 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

RAG Evaluation Harnesses, Hallucination Guardrails & Hybrid Search

Self-explanatory list-and-bullet production mastery module covering RAG Evaluation Harnesses, Hallucination Guardrails & Hybrid Search in Applied Data Science, GenAI & MLOps Engineering, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken MockAttempt AI Viva.

Objectives: ✓ Chunking Strategies, Metadata Filtering & Hybrid Keyword + Vector Retrieval — Purpose: Retrieve exact enterprise policy clauses with high recall instead of noisy semantic-only matches. • ✓ Cross-Encoder Re-Ranking, Citation Enforcement & Prompt-Injection Guards — Purpose: Block jailbreak attempts and force LLM responses to cite verifiable source passages. • ✓ Automated RAG Evaluation (Faithfulness, Context Precision & Latency Budgets) — Purpose: Catch prompt regressions and hallucination spikes automatically before shipping model updates. • ✓ Build, verify, and document 'Grounded Hybrid BM25 + pgvector RAG Pipeline with Automated Faithfulness Scoring' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + MockAttempt AI Technical Interviews.

13.1 Chunking Strategies, Metadata Filtering & Hybrid Keyword + Vector Retrieval [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Chunking Strategies, Metadata Filtering & Hybrid Keyword + Vector Retrieva... • Performance Optimization, Complexity & Latency Impact of Chunking Strategies, Metadata Filtering & Hybrid Keyword +... • Live Debugging, Stack • Runnable Code Implementation (PYTHON) • Hands • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Chunking Strategies • Production Hardening & Security: OAuth2 scopes, prompt-injection guardrails & token-rate quotas

Purpose: To understand and apply Chunking Strategies, Metadata Filtering & Hybrid Keyword + Vector Retrieval in real Applied Data Science, GenAI & MLOps Engineering systems — spec...

What Topic Is Next → Next Topic 13.2: Cross-Encoder Re-Ranking, Citation Enforcement & Prompt-Injection Guards

13.2 Cross-Encoder Re-Ranking, Citation Enforcement & Prompt-Injection Guards [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Cross • Performance Optimization, Complexity & Latency Impact of Cross • Live Debugging, Stack • Runnable Code Implementation (PYTHON) • Hands • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Cross-Encoder Re-Ranking • Production Hardening & Security: OAuth2 scopes, prompt-injection guardrails & token-rate quotas

Purpose: To understand and apply Cross-Encoder Re-Ranking, Citation Enforcement & Prompt-Injection Guards in real Applied Data Science, GenAI & MLOps Engineering systems — specifi...

What Topic Is Next → Next Topic 13.3: Automated RAG Evaluation (Faithfulness, Context Precision & Latency Budgets)

13.3 Automated RAG Evaluation (Faithfulness, Context Precision & Latency Budgets) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Automated RAG Evaluation (Faithfulness, Context Precision & Latency Budget... • Performance Optimization, Complexity & Latency Impact of Automated RAG Evaluation (Faithfulness, Context Precision.... • Live Debugging, Stack • Runnable Code Implementation (PYTHON) • Hands • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Automated RAG Evaluation (Faithfulness, Context Prec... • Production Hardening & Security: OAuth2 scopes, prompt-injection guardrails & token-rate quotas

Purpose: To understand and apply Automated RAG Evaluation (Faithfulness, Context Precision & Latency Budgets) in real Applied Data Science, GenAI & MLOps Engineering systems — spe...

What Topic Is Next → Module 13 Capstone (Grounded Hybrid BM25 + pgvector RAG Pipeline with Automated Faithfulness Scoring) → Next Topic 14.1: Feature Skew Detection, Data Drift Alerts (PSI/KS) & Shadow Deployments

Module 13 Hands-On Capstone Project (100 Marks): Grounded Hybrid BM25 + pgvector RAG Pipeline with Automated Faithfulness Scoring — Architect, implement, test, and document 'Grounded Hybrid BM25 + pgvector RAG Pipeline with Automated Faithfulness Scoring' applying every bullet-wise engineering concept from Module 13 (RAG...

MODULE 14 OF 14 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Production MLOps, Model Drift Monitoring & AI System Design

Self-explanatory list-and-bullet production mastery module covering Production MLOps, Model Drift Monitoring & AI System Design in Applied Data Science, GenAI & MLOps Engineering, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken MockAttempt AI Viva.

Objectives: ✓ Feature Skew Detection, Data Drift Alerts (PSI/KS) & Shadow Deployments — Purpose: Detect when live customer distributions diverge from training data before prediction accuracy degrades. • ✓ Async Batching, KV-Cache / Semantic Caching & Token Cost Governance — Purpose: Cut LLM and ML inference latency and cloud GPU/API bills by up to 60% in production. • ✓ End-to-End ML & GenAI System Design Case Studies for Product Interviews — Purpose: Design recommendation, fraud-scoring, and enterprise copilot architectures step-by-step. • ✓ Build, verify, and document 'Canary Model Serving Gateway, Drift Alerting & Token-Cost Telemetry Service' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + MockAttempt AI Technical Interviews.

14.1 Feature Skew Detection, Data Drift Alerts (PSI/KS) & Shadow Deployments [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Feature Skew Detection, Data Drift Alerts (PSI/KS) & Shadow Deployments • Performance Optimization, Complexity & Latency Impact of Feature Skew Detection, Data Drift Alerts (PSI/KS) & Shado... • Live Debugging, Stack • Runnable Code Implementation (PYTHON) • Hands • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Feature Skew Detection • Production Hardening & Security: OAuth2 scopes, prompt-injection guardrails & token-rate quotas

Purpose: To understand and apply Feature Skew Detection, Data Drift Alerts (PSI/KS) & Shadow Deployments in real Applied Data Science, GenAI & MLOps Engineering systems — specific...

What Topic Is Next → Next Topic 14.2: Async Batching, KV-Cache / Semantic Caching & Token Cost Governance

14.2 Async Batching, KV-Cache / Semantic Caching & Token Cost Governance [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Async Batching, KV • Performance Optimization, Complexity & Latency Impact of Async Batching, KV • Live Debugging, Stack • Runnable Code Implementation (PYTHON) • Hands • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for Async Batching • Production Hardening & Security: OAuth2 scopes, prompt-injection guardrails & token-rate quotas

Purpose: To understand and apply Async Batching, KV-Cache / Semantic Caching & Token Cost Governance in real Applied Data Science, GenAI & MLOps Engineering systems — specifically...

What Topic Is Next → Next Topic 14.3: End-to-End ML & GenAI System Design Case Studies for Product Interviews

14.3 End-to-End ML & GenAI System Design Case Studies for Product Interviews [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for End • Performance Optimization, Complexity & Latency Impact of End • Live Debugging, Stack • Runnable Code Implementation (PYTHON) • Hands • Idiomatic DSAI Implementation: async/await non-blocking endpoints, typed Pydantic contracts & grounded RAG • Asyncio Event Loop, FastAPI ASGI, Pydantic v2 & pgvector/PyTorch Execution & Memory Lifecycle for End-to-End ML • Production Hardening & Security: OAuth2 scopes, prompt-injection guardrails & token-rate quotas

Purpose: To understand and apply End-to-End ML & GenAI System Design Case Studies for Product Interviews in real Applied Data Science, GenAI & MLOps Engineering systems — specific...

What Topic Is Next → Module 14 Capstone → End-of-Course Live Client Project: DocuSense Enterprise GenAI RAG Knowledge Copilot & Predictive Scoring API

Module 14 Hands-On Capstone Project (100 Marks): Canary Model Serving Gateway, Drift Alerting & Token-Cost Telemetry Service — Architect, implement, test, and document 'Canary Model Serving Gateway, Drift Alerting & Token-Cost Telemetry Service' applying every bullet-wise engineering concept from Module 14 (Producti...

3. LIVE CLIENT PRODUCTION PROJECT & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: DocuSense Enterprise GenAI RAG Knowledge Copilot & Predictive Scoring API • **Confirmed Role:** AI, Data Science & GenAI MLOps Intern

Domain & Production Infra: Healthcare & Enterprise Knowledge Operations Client (Python + FastAPI + PostgreSQL pgvector + Cloud GPU/CPU Inference Endpoint)

- Built automated Python feature-engineering pipelines and calibrated predictive ML models with Scikit-Learn & PyTorch.
- Engineered a grounded Retrieval-Augmented Generation (RAG) service over PostgreSQL pgvector with citation verification.
- Exposed low-latency async FastAPI inference endpoints with token streaming, guardrails, and structured evaluation metrics.
- Deployed model-serving containers to live client production with drift monitoring and automated regression tests.

Target Job Roles: Applied AI & GenAI Engineer, Data Scientist / Machine Learning Engineer, LLM RAG & MLOps Engineer, Predictive Analytics & AI Developer

Realistic Annual CTC Scope: **Fresher / Associate Data Scientist & AI Engineer (0–1 Yr):** ₹6.5 LPA – ₹13.5 LPA **Mid-Level ML / GenAI Engineer (1–3 Yrs):** ₹14.0 LPA – ₹28.0 LPA **Senior AI Architect / Principal ML Engineer (3–6+ Yrs):** ₹28.0 LPA – ₹58.0+ LPA

MockAttempt Academy Pvt. Ltd. • Email: support@mockattempt.com • Phone/WhatsApp: +91 95400 07917 • Web: https://mockattempt.com/courses/applied-data-science-genai-mlops-engineering

Try 2 Free Topic-Wise Timed Topic Mock Tests & Spoken MockAttempt AI Interviews daily on every topic above before enrolling.