Enterprise AI & LLM Engineering Capability Program

A 10-Day Customized, Hands-On Training Path

Table of Contents & Curriculum Outline

Program Approach

This program is built for developers who already know Python and R and need to move directly into practical, enterprise-grade AI engineering — not introductory theory. It is structured as a four-phase, project-based path: foundations and large-scale data handling, applied machine learning and NLP, LLM fine-tuning and Retrieval-Augmented Generation, and finally deployment, monitoring, and governance. Every day pairs a concise concept session with a hands-on lab built around NSSF-style scenarios such as claims risk, fraud detection, contribution forecasting, and member service automation.

Technology stance: Open-source and vendor-neutral first (PyTorch, HuggingFace, LangChain, FAISS/Chroma, MLflow), with a dedicated look at deployment patterns on Azure, AWS, and Google Cloud so the team understands the trade-offs before committing to any platform.

Delivery model: Each day is split into a morning concept/design session and an afternoon hands-on lab. Labs use synthetic, NSSF-shaped datasets (contributions, claims, member records, feedback) so skills transfer directly to real systems without exposing production data during training.

Program Structure at a Glance

Phase	Days	Focus
Phase 1	1–3	Enterprise AI foundations, applied ML, deep learning, and large-scale data engineering
Phase 2	4–6	Predictive analytics, forecasting, fraud/anomaly detection, and NLP/sentiment analysis
Phase 3	7–9	LLM foundations, fine-tuning open-source models, and Retrieval-Augmented Generation
Phase 4	10	Deployment, monitoring, governance, and a capstone project presentation

PHASE 1 — FOUNDATIONS & DATA ENGINEERING

Day 1: Enterprise AI Foundations & Open-Source Tooling Setup

LEARNING OBJECTIVES

- Align the training path with NSSF's digital transformation objectives and identify priority use cases
- Compare open-source and cloud-based AI architectures and understand when each applies
- Stand up a shared, reproducible open-source development environment

MORNING SESSION

- The enterprise AI landscape for public-sector and financial institutions
- Open-source vs. cloud-native AI architectures: components, trade-offs, and vendor-neutral design principles
- Mapping NSSF priorities — fraud detection, forecasting, service delivery, decision support — to AI technique categories

AFTERNOON SESSION — HANDS-ON LAB

- Lab: set up Python environments (venv/conda), Git workflow, and a shared Jupyter/VS Code workspace
- Lab: exploratory data analysis on a synthetic NSSF-style contributions/claims dataset using pandas and numpy
- Lab: first end-to-end mini-pipeline — load, clean, visualize, and profile a large tabular dataset

AT A GLANCE

Focus Area	Environment setup, data landscape, architecture options
Tools & Tech	Python, Jupyter/VS Code, Git, pandas, numpy, scikit-learn
NSSF Use Case	Establishing a reproducible foundation for all later NSSF data-science work

PHASE 1 — FOUNDATIONS & DATA ENGINEERING

Day 2: Machine Learning Foundations for Enterprise Applications

LEARNING OBJECTIVES

- Apply supervised and unsupervised learning to enterprise decision problems
- Engineer features from operational data and evaluate models rigorously
- Build a first classification and clustering model on NSSF-style data

MORNING SESSION

- Supervised learning: regression and classification algorithms (logistic regression, decision trees, ensembles)
- Unsupervised learning: clustering and segmentation techniques
- Feature engineering, cross-validation, and evaluation metrics (precision/recall, ROC-AUC, RMSE)

AFTERNOON SESSION — HANDS-ON LAB

- Lab: build a claims risk-scoring classifier with scikit-learn and XGBoost
- Lab: build a member/employer segmentation model using k-means and hierarchical clustering
- Lab: compare models with a shared evaluation harness and interpret results with feature importance

AT A GLANCE

Focus Area	Supervised & unsupervised ML, evaluation
Tools & Tech	scikit-learn, XGBoost, LightGBM, matplotlib/seaborn
NSSF Use Case	Risk-scoring claims and segmenting members/employers for targeted service delivery

PHASE 1 — FOUNDATIONS & DATA ENGINEERING

Day 3: Deep Learning & Working with Large-Scale Enterprise Data

LEARNING OBJECTIVES

- Understand neural network fundamentals and when deep learning adds value over classical ML
- Handle enterprise-scale datasets (millions of records) efficiently
- Build and version reproducible data pipelines

MORNING SESSION

- Neural network fundamentals: architecture, backpropagation, training dynamics
- Overview of CNNs and RNNs and where they fit in enterprise use cases
- Scaling beyond a single machine: distributed processing with Spark/Dask, and data versioning with DVC

AFTERNOON SESSION — HANDS-ON LAB

- Lab: build and train a simple feed-forward network in PyTorch
- Lab: process a large synthetic contributions dataset (millions of rows) with Spark or Dask
- Lab: design a versioned ETL pipeline for AI-ready data using DVC

AT A GLANCE

Focus Area	Deep learning basics, large-scale data engineering
Tools & Tech	PyTorch, TensorFlow (overview), Apache Spark/Dask, DVC
NSSF Use Case	Efficiently processing and versioning NSSF's full member and contributions history

PHASE 2 — APPLIED ML, FORECASTING & NLP

Day 4: Predictive Analytics & Forecasting

LEARNING OBJECTIVES

- Build time-series forecasting models for financial and operational planning
- Apply proper backtesting and validation for forecasting models
- Translate forecasts into decision-support outputs

MORNING SESSION

- Time-series concepts: trend, seasonality, stationarity
- Forecasting methods: ARIMA, Prophet, and gradient-boosted models for time series
- Backtesting strategies and forecast uncertainty/confidence intervals

AFTERNOON SESSION — HANDS-ON LAB

- Lab: forecast monthly contribution inflows using Prophet and statsmodels
- Lab: forecast benefit payout trends with XGBoost-based time-series features
- Lab: build a backtesting harness and present forecasts with confidence bands

AT A GLANCE

Focus Area	Time-series forecasting for planning
Tools & Tech	statsmodels, Prophet, XGBoost, pandas
NSSF Use Case	Forecasting contribution inflows and benefit payouts for financial planning

PHASE 2 — APPLIED ML, FORECASTING & NLP

Day 5: Fraud & Anomaly Detection with Machine Learning

LEARNING OBJECTIVES

- Apply anomaly-detection techniques to identify irregular transactions or claims
- Handle severely imbalanced datasets correctly
- Design a fraud-scoring workflow suitable for production review

MORNING SESSION

- Anomaly detection techniques: isolation forests, one-class SVM, autoencoders
- Handling imbalanced data: SMOTE and cost-sensitive learning
- Designing human-in-the-loop review workflows for flagged cases

AFTERNOON SESSION — HANDS-ON LAB

- Lab: build an isolation-forest and autoencoder-based fraud detector on synthetic claims/transaction data
- Lab: rebalance training data with SMOTE and compare model performance
- Lab: design a scoring and case-review workflow with configurable thresholds

AT A GLANCE

Focus Area	Fraud detection, anomaly detection, imbalanced data
Tools & Tech	scikit-learn, PyOD, imbalanced-learn, PyTorch (autoencoders)
NSSF Use Case	Flagging suspicious claims, registrations, or contribution patterns for review

PHASE 2 — APPLIED ML, FORECASTING & NLP

Day 6: Natural Language Processing & Sentiment Analysis

LEARNING OBJECTIVES

- Process and analyze unstructured text at scale
- Build sentiment-analysis and entity-extraction models
- Understand how classical NLP connects to modern transformer-based NLP

MORNING SESSION

- Text preprocessing and classical NLP: tokenization, embeddings (Word2Vec/GloVe)
- Introduction to transformer-based NLP (BERT-family models) for classification and extraction
- Named Entity Recognition (NER) and topic modeling for unstructured records

AFTERNOON SESSION — HANDS-ON LAB

- Lab: build a sentiment-analysis model on synthetic member feedback/complaints data
- Lab: fine-tune a small BERT-family model for complaint categorization

- Lab: extract structured entities (names, dates, amounts) from free-text records with spaCy

AT A GLANCE

Focus Area	NLP, sentiment analysis, NER
Tools & Tech	spaCy, NLTK, HuggingFace Transformers, scikit-learn
NSSF Use Case	Analyzing member feedback and complaints to surface service issues automatically

PHASE 3 — LLM ENGINEERING & RAG

Day 7: Introduction to LLMs, Transformer Architecture & the Foundation-Model Lifecycle

LEARNING OBJECTIVES

- Understand the transformer architecture underlying modern LLMs
- Survey the landscape of open-source LLMs suitable for enterprise deployment
- Gain a high-level view of how foundation models are built from scratch — data, infrastructure, and cost — to inform future architectural decisions

MORNING SESSION

- Transformer architecture: attention, embeddings, tokenization
- Landscape of open-source LLMs (e.g., Llama, Mistral, Falcon families) and how to evaluate them for a use case
- End-to-end foundation-model lifecycle overview: pretraining data, compute infrastructure, cost, and practical challenges (context-setting only, not a build exercise)

AFTERNOON SESSION — HANDS-ON LAB

- Lab: run and query open-source LLMs locally/on-prem using Ollama and HuggingFace
- Lab: structured prompt-engineering exercises for enterprise tasks (summarization, classification, Q&A)
- Lab: build a model-selection scorecard comparing candidate open-source LLMs against NSSF requirements (data sensitivity, size, licensing, performance)

AT A GLANCE

Focus Area	Transformers, open-source LLM landscape, foundation-model lifecycle
Tools & Tech	HuggingFace Transformers, Ollama, LangChain
NSSF Use Case	Selecting a shortlist of open-source LLMs appropriate for NSSF's data-sensitive environment

PHASE 3 — LLM ENGINEERING & RAG

Day 8: Fine-Tuning & Customizing Open-Source LLMs

LEARNING OBJECTIVES

- Prepare organizational data for LLM fine-tuning
- Apply parameter-efficient fine-tuning (PEFT) techniques to adapt an open-source LLM
- Evaluate a fine-tuned model against a baseline

MORNING SESSION

- Fine-tuning strategies: full fine-tuning vs. parameter-efficient methods (LoRA, QLoRA)
- Instruction tuning and dataset preparation from organizational documents/FAQs
- Evaluation of fine-tuned LLMs: accuracy, faithfulness, and regression checks against the base model

AFTERNOON SESSION — HANDS-ON LAB

- Lab: prepare an instruction-tuning dataset from sample NSSF policy/FAQ documents
- Lab: fine-tune an open-source LLM (e.g., Llama or Mistral family) using LoRA/QLoRA
- Lab: evaluate the fine-tuned model against the base model on held-out policy questions

AT A GLANCE

Focus Area	PEFT fine-tuning (LoRA/QLoRA), instruction tuning
Tools & Tech	HuggingFace PEFT & Trainer, bitsandbytes, transformers
NSSF Use Case	Adapting an LLM to answer NSSF policy and member-service questions accurately

PHASE 3 — LLM ENGINEERING & RAG

Day 9: Retrieval-Augmented Generation (RAG) & Enterprise Integration

LEARNING OBJECTIVES

- Design and build a Retrieval-Augmented Generation pipeline over organizational documents
- Integrate an LLM/RAG service into an existing in-house application via an API
- Understand cloud deployment options for LLM workloads as a point of comparison

MORNING SESSION

- RAG architecture: chunking, embeddings, vector databases, retrieval strategies
- Integration patterns: exposing AI services via internal APIs, authentication, and rate limiting
- Overview of managed LLM deployment options on Azure OpenAI, AWS Bedrock, and Google Vertex AI, and how they compare to a self-hosted open-source stack

AFTERNOON SESSION — HANDS-ON LAB

- Lab: build a full RAG pipeline over sample NSSF policy/member-handbook documents with a vector database
- Lab: wrap the RAG pipeline in a FastAPI service and call it from a sample in-house application
- Lab: walkthrough of deploying the same pipeline on one major cloud platform, compared side-by-side with the self-hosted version

AT A GLANCE

Focus Area	RAG pipelines, API integration, cloud deployment overview
Tools & Tech	LangChain/LlamaIndex, FAISS/Chroma, FastAPI, Azure/AWS/GCP (overview)
NSSF Use Case	An internal knowledge-assistant that answers member and staff questions from NSSF policy documents

PHASE 4 — DEPLOYMENT, GOVERNANCE & CAPSTONE

Day 10: AI Model Deployment, Monitoring, Governance & Capstone Project

LEARNING OBJECTIVES

- Package and deploy AI/LLM services using enterprise-grade MLOps practices
- Monitor models in production for drift, performance, and cost
- Apply governance, security, and responsible-AI practices to deployed systems
- Present a complete, integrated solution to an NSSF use case

MORNING SESSION

- Deployment: containerization with Docker, orchestration basics with Kubernetes, CI/CD for ML/LLM services
- Monitoring: model and data drift detection, logging, latency/cost tracking
- Governance: data privacy, access control, model documentation, bias and responsible-AI review, and an internal AI governance checklist

AFTERNOON SESSION — HANDS-ON LAB

- Capstone: teams package and deploy one end-to-end solution built during the week (fraud detector, forecasting model, or RAG assistant) behind a monitored API
- Capstone presentations: each team demonstrates their solution, walks through its architecture, and discusses governance considerations
- Closing session: roadmap for continued internal capability-building and a prioritized backlog of next AI use cases for NSSF

AT A GLANCE

Focus Area	MLOps, monitoring, governance, capstone delivery
Tools & Tech	Docker, Kubernetes (overview), MLflow, Prometheus/Grafana
NSSF Use Case	A governed, monitored deployment pipeline plus a working prototype for a real NSSF use case