

AI Training for Pension Funds, Social Security and Insurance

From member, contribution and claims data to secure, production-ready AI: machine learning, deep learning, NLP and self-hosted GenAI built on your institution's own synthetic data.

Metadata

Item	Detail
Duration	10 days, 80 hours
Audience	Technical teams in data, risk, IT and digital functions at pension funds, social security institutions and insurance companies
Experience level	Beginner in AI/ML; technically minded with an understanding of the business
Hands-on ratio	70%
Learning environment	Koenig DC with GPU (preferred), or client GPU-enabled machines
Capstone	Optional, if opted for by the client

Overview

This 10-day programme takes technical teams at pension funds, social security institutions and insurers from Python fundamentals to a secured, production-ready AI system. Every lab runs on one synthetic database of members, employers, contributions, policies, claims and benefit payments, modelled on the client's own schemes and products, so each day builds on the last.

Learners apply machine learning to contribution arrears and policy lapse, claim cost, claims and benefit fraud, and contribution and benefit cash-flow forecasting. They then move into neural networks, multilingual NLP and self-hosted GenAI, working on claims documents, member complaints, beneficiary and KYC documents and a local-language member assistant. The programme ends with security hardening and deployment aligned to pension, insurance and social security regulators and data-protection requirements.

By the end of the course, learners will be able to:

- Explore and assess member, contribution and claims data quality with Python
- Build, evaluate and explain ML models for arrears, lapse, claims and fraud use cases
- Forecast contribution inflows and benefit outflows
- Train neural networks on GPU for tabular, sequence and anomaly problems
- Process multilingual claims and member text for classification and entity extraction

- Host, prompt, fine-tune and ground an LLM with retrieval-augmented generation
- Defend GenAI systems against prompt injection and data leakage
- Harden and ship AI services to a compliant production environment

Prerequisites Required

- Technical background in data, risk, IT or digital roles
- Working understanding of pension, social security or insurance products and processes, such as contributions, benefits, policies and claims
- Prior Python or machine learning experience is not required; Day 1 covers Python basics

Tools & Technology Required

Learning Environment

- Trainer recommends Koenig DC for a standardised learning environment with GPU access.
- Learners must be informed at least 10 days before the course begins.
- If the client has GPU-enabled machines, delivery begins on those instead.

Programming

- Python, used from Day 1 onwards
- Python data, machine learning and deep learning libraries; trainer to confirm the list and versions

Data

- Synthetic database of members, employers, contributions, policies, claims and benefit payments, modelled on the client's schemes and products
- Database supplied by the client or generated by the trainer, and reused across all 10 days
- Synthetic multilingual document set, including claims documents, member complaints, beneficiary and KYC documents, and scheme rules

Languages for NLP and GenAI Labs

NLP and GenAI tasks are grounded in the client's own language material. Languages are selected per client from:

- African languages: Swahili, Hausa, Amharic, Yoruba, Oromo, Igbo, Fula, Lingala, Zulu and Somali
- Arabic, English, Spanish and Portuguese
- Chinese, Taiwanese, Korean, Hindi and Tamil
- Other major languages as required by the client

GenAI

- Self-hosted large language model running on GPU within the lab environment
- Model choice and serving stack to be confirmed by trainer

Pedagogy

- Instructor demonstrations introduce each new technique
- Guided labs walk learners through each build step by step
- Hands-on activities make up 70% of the course
- Qubits (MCQ checkpoints) test and reinforce learning each day
- Discussions connect each technique to pension, social security and insurance practice
- One synthetic scheme case runs through all 10 days
- Capstone project if opted for by the client; learning is then driven by the project itself, with the basics covered in the lab environment

Course Schedule

Day	Topics Covered	Hands-on Activities	Tools & Technology Required
1	Python basics; synthetic scheme data; exploratory data analysis	Environment setup; data quality report; member, contribution and claims EDA	Koenig DC, Python, synthetic database
2	ML fundamentals; supervised learning; evaluation metrics	Feature engineering; baseline arrears and claim cost models	Koenig DC, Python
3	Ensembles; explainability; survival analysis; anomaly detection; forecasting	Tuned lapse model; longevity curves; claims and benefit fraud detector; cash-flow forecast	Koenig DC, Python
4	Neural network fundamentals; GPU training	Neural network lapse model benchmarked against Day 3	Koenig DC (GPU), Python
5	Sequence models; autoencoders; attention	Sequence-based cash-flow forecast; autoencoder fraud detector	Koenig DC (GPU), Python
6	Multilingual NLP; transformers	Complaint triage classifier; claims and KYC entity extraction	Koenig DC (GPU), Python, multilingual documents
7	Self-hosted LLMs; prompt engineering; prompt injection	Host LLM; build member assistant prompts; red-team and mitigate	Koenig DC (GPU), self-hosted LLM
8	Fine-tuning strategy; parameter-efficient fine-tuning	Fine-tune a local-language member assistant	Koenig DC (GPU), self-hosted LLM
9	Retrieval-augmented generation	Grounded member assistant over scheme rules and policy wordings	Koenig DC (GPU), self-hosted LLM
10	Security, hardening, MLOps; production deployment	Harden, package and deploy AI services; security review	Koenig DC (GPU), self-hosted LLM

Detailed Day-wise Flow

Day 1: Python Foundations and Scheme Data Exploration

Learning Outcomes

- Write Python to load, clean and summarise member, contribution and claims data
- Profile the synthetic scheme database and judge its fitness for modelling
- Identify patterns and risks in members, contributions, claims and benefit payments

Topics Covered

- AI and ML use cases across pension funds, social security institutions and insurers
- Synthetic data: purpose, privacy benefits and fidelity to client schemes and products
- Python essentials: data types, control flow, functions and modules
- Working with tabular data: dataframes, joins and aggregations
- Querying the synthetic database from Python
- Data quality: missing values, duplicates, outliers and inconsistent member records
- Exploratory analysis: member demographics, contribution regularity, claims experience and benefit payments
- Visualising scheme data for business stakeholders

Hands-on Activities

- Set up and validate the Python environment in Koenig DC
- Explore the schema of the synthetic member, employer, contribution, policy, claims and benefit tables
- Produce a data quality report for the synthetic database
- Analyse contribution regularity and employer arrears by segment
- Analyse claims frequency, cost and benefit payment patterns
- Qubit: Python and data exploration checkpoint

Capstone Progress (if opted for by the client)

- Select the capstone business scenario and profile its data

Day 2: Machine Learning Fundamentals

Learning Outcomes

- Frame pension, social security and insurance problems as machine learning tasks
- Engineer features from member, contribution and claims data
- Build and evaluate baseline classification and regression models

Topics Covered

- The ML workflow from problem framing to evaluation
- Supervised and unsupervised learning in a scheme and insurance context
- Train, validation and test splits; time-based splits for contribution and claims data
- Data leakage and how it arises in member and claims datasets
- Feature engineering: contribution history, tenure, claims history and demographic features
- Regression and classification; logistic regression and decision trees
- Evaluation metrics: ROC-AUC, precision, recall, confusion matrix, MAE and calibration
- Handling class imbalance in arrears, lapse and fraud data

Hands-on Activities

- Build a reusable feature engineering pipeline on the synthetic database
- Train a baseline model to predict employer contribution arrears
- Train a baseline regression model to estimate claim cost
- Evaluate models with business-relevant metrics and thresholds
- Qubit: ML fundamentals checkpoint

Capstone Progress (if opted for by the client)

- Build the baseline model for the capstone scenario

Day 3: Applied Machine Learning for Lapse, Longevity, Fraud and Cash Flow**Learning Outcomes**

- Improve predictive models with ensembles and tuning
- Explain model decisions in terms regulators and actuarial committees accept
- Model time-to-event outcomes, detect fraud and forecast cash flows

Topics Covered

- Ensemble methods: random forests and gradient boosting
- Cross-validation and hyperparameter tuning
- Model explainability: feature importance and per-decision explanations
- Bias, fairness and model validation documentation
- Survival analysis for longevity, policy lapse and time to claim
- Anomaly detection for claims fraud and benefit fraud, such as duplicate claims and payments to deceased beneficiaries
- Time-series fundamentals for contribution inflows and benefit outflows

Hands-on Activities

- Tune a gradient-boosted policy lapse model and compare it with a logistic regression baseline
- Generate explanations for individual lapse and arrears predictions
- Build survival curves for member longevity and policy lapse
- Build an anomaly detector to flag suspicious claims and benefit payments
- Forecast monthly contribution inflows and benefit outflows from the synthetic data
- Qubit: applied ML checkpoint

Capstone Progress (if opted for by the client)

- Improve and explain the capstone model

Day 4: Neural Network Foundations**Learning Outcomes**

- Explain how neural networks learn from data
- Train neural networks on GPU in Koenig DC
- Judge when a neural network outperforms tree-based models on scheme and claims data

Topics Covered

- Neurons, layers, activation functions and network architecture
- Loss functions, backpropagation and gradient descent
- Optimisers, learning rates and batch training
- Overfitting control: regularisation, dropout and early stopping
- GPU acceleration and efficient training practices
- Neural networks for tabular member and claims data

Hands-on Activities

- Build and train a multi-layer network on the synthetic policy lapse data
- Tune architecture and regularisation to control overfitting
- Benchmark the network against the Day 3 gradient-boosted model
- Qubit: neural network fundamentals checkpoint

Capstone Progress (if opted for by the client)

- Add a neural network variant to the capstone model and benchmark it

Day 5: Advanced Neural Networks for Sequences and Anomalies**Learning Outcomes**

- Model contribution histories and cash flows with neural networks
- Detect unusual claims and benefit payments with reconstruction-based anomaly detection
- Describe how attention leads to transformer models

Topics Covered

- Sequence data in schemes and insurance: contribution histories, claims histories and cash flows
- Recurrent networks and LSTMs
- Embeddings for members, employers, providers and categorical features
- Autoencoders for anomaly detection
- Attention and the transformer architecture

Hands-on Activities

- Forecast contribution inflows and benefit outflows with a sequence model and compare with Day 3
- Build an autoencoder to flag anomalous claims and benefit payments
- Compare autoencoder alerts with the Day 3 fraud detector
- Qubit: sequence and anomaly models checkpoint

Capstone Progress (if opted for by the client)

- Add a sequence or anomaly component to the capstone solution

Day 6: Multilingual NLP for Pensions, Social Security and Insurance

Learning Outcomes

- Prepare multilingual member and claims text for analysis
- Classify member complaints across languages
- Extract structured information from claims, beneficiary and KYC documents

Topics Covered

- NLP use cases: claims documents, member complaints, beneficiary records and regulatory text
- Text preprocessing across scripts and languages
- Tokenisation and challenges for low-resource African languages
- Word and sentence embeddings
- Pretrained and multilingual transformer models
- Text classification, sentiment analysis and named entity recognition
- Handling personal and health data in text

Hands-on Activities

- Explore the synthetic multilingual complaints, claims and KYC document set
- Build a complaint triage classifier by scheme or product, issue and urgency
- Extract names, member numbers, dates and amounts from synthetic claims and KYC documents
- Evaluate model performance by language and identify gaps
- Qubit: NLP checkpoint

Capstone Progress (if opted for by the client)

- Add a text analytics component in the client's language set

Day 7: Self-hosted LLMs, Prompt Engineering and Prompt Injection

Learning Outcomes

- Run an open-weight LLM on the institution's own GPU infrastructure
- Design reliable prompts for member service and claims tasks
- Attack and defend an LLM application against prompt injection

Topics Covered

- Why pension funds, social security institutions and insurers self-host: data residency, confidentiality and control
- Open-weight model selection, model size and quantisation
- Serving an LLM on GPU and calling it from applications
- Prompt engineering: instructions, few-shot examples, structured output and reasoning steps
- System prompts and role design for a member assistant
- Prompt injection: direct, indirect and jailbreak attacks
- Data leakage risks and guardrail patterns

Hands-on Activities

- Trainer demonstration: deploying and serving the self-hosted LLM in Koenig DC
- Learners connect to the hosted model and test its multilingual behaviour
- Build prompts that summarise and route claims and member complaints
- Draft the system prompt for the local-language member assistant
- Red-team the assistant with injection attacks and apply mitigations
- Qubit: prompting and LLM security checkpoint

Capstone Progress (if opted for by the client)

- Design and red-team the prompts for the capstone assistant

Day 8: Fine-tuning AI Models

Learning Outcomes

- Decide when fine-tuning beats prompting or retrieval
- Prepare an instruction dataset from synthetic scheme and claims material
- Fine-tune and evaluate a model for a local-language member assistant

Topics Covered

- Prompting versus fine-tuning versus retrieval: a decision framework
- Instruction datasets: format, quality, balance and personal-data scrubbing
- Full fine-tuning versus parameter-efficient methods such as LoRA and QLoRA
- Fine-tuning for low-resource and local languages
- Evaluation: task accuracy, language quality and regression checks
- Versioning and documenting fine-tuned models

Hands-on Activities

- Build an instruction dataset of member queries on contributions, benefits and claims in the client's selected languages
- Fine-tune a model with a parameter-efficient method on GPU
- Compare the base and fine-tuned model on member service tasks by language
- Qubit: fine-tuning checkpoint

Capstone Progress (if opted for by the client)

- Fine-tune the model component of the capstone solution

Day 9: Retrieval-Augmented Generation

Learning Outcomes

- Design a RAG pipeline over scheme rules, policy wordings and benefit regulations
- Ground assistant answers in approved sources with citations
- Measure the accuracy and faithfulness of RAG responses

Topics Covered

- RAG architecture: ingestion, chunking, embedding, retrieval and generation
- Chunking strategies for scheme rules, policy wordings and claims guidance
- Multilingual embeddings and cross-language retrieval
- Vector stores, metadata filtering and document-level access control
- Reranking, citations and refusal when no source applies
- RAG evaluation: retrieval quality, faithfulness and answer relevance

Hands-on Activities

- Ingest synthetic scheme rules, policy wordings and claims guidance in multiple languages
- Build a retrieval pipeline and connect it to the self-hosted LLM
- Add citations and safe refusal behaviour to the member assistant
- Evaluate RAG answers against a test question set and tune retrieval
- Qubit: RAG checkpoint

Capstone Progress (if opted for by the client)

- Integrate retrieval into the capstone assistant

Day 10: Security, Hardening and Shipping to Production

Learning Outcomes

- Threat-model ML and GenAI systems in a regulated pension, social security or insurance institution
- Apply controls that meet regulator and data-protection requirements
- Package, deploy and monitor AI services in a production-like environment

Topics Covered

- Threats to ML and LLM systems, including the OWASP Top 10 for LLM applications
- Data protection: personal and health data masking, encryption, access control and data residency
- Model risk management: validation, documentation, approvals and audit trails
- Self-hosted versus cloud deployment for regulated institutions
- Packaging models and LLM services as deployable APIs
- Monitoring: data drift, model performance, LLM output quality and abuse
- Incident response and rollback for AI systems

Hands-on Activities

- Threat-model the fraud detector and the member assistant
- Add personal-data redaction, input and output filtering, and audit logging
- Package the lapse model and RAG assistant as services
- Deploy the services to a production-like environment in Koenig DC
- Run a security and compliance checklist against the deployment
- Qubit: security and production checkpoint

Capstone Progress (if opted for by the client)

- Harden, deploy and present the capstone solution