Artificial Intelligence & Machine Learning Developer Program

Course Overview

Master the fundamentals and advanced concepts of Artificial Intelligence (AI), Machine Learning (ML), Python Programming, Data Analysis, Statistics, Deep Learning, Neural Networks, Natural Language Processing (NLP), Computer Vision, Generative AI, and Large Language Models (LLMs) with Tech Learniversity's Artificial Intelligence & Machine Learning Developer Professional Training Program.

This comprehensive beginner-to-advanced program combines live instructor-led training, hands-on Python coding exercises, practical Machine Learning experiments, model evaluation, and real-world AI application development to help learners analyze data, build predictive models, develop intelligent applications, and understand the fundamentals of deploying AI-powered solutions.

Course Duration: 180 Hours of Live Instructor-Led Training, including 72 Hours of Concepts and Guided Discussion and 108 Hours of Practical Labs and Project Work.

Course Level: Beginner to Advanced — Designed for Students, Graduates, Freshers, Technical Professionals, Aspiring AI Developers, and Career Changers. No Prior Python Programming or Artificial Intelligence Experience Required.

Course Structure: 17 Comprehensive Modules with Guided Coding Exercises, Practical Assessments, Machine Learning Experiments, AI Application Development, and an Integrated Final Capstone Project.

Practical Projects: Hands-On AI and Machine Learning Projects covering House Price Prediction, Customer Churn Prediction, Customer Segmentation, Spam Detection or Sentiment Analysis, Image Classification, Sales Forecasting, and AI Document Question-Answering using Retrieval-Augmented Generation (RAG). Learners select one project or an instructor-approved equivalent for the 22-hour Integrated Final Capstone.

Tools & Technologies: Python, Google Colab, Jupyter Notebook, NumPy, Pandas, SQL, SQLite, scikit-learn, TensorFlow, Keras, Hugging Face Transformers, Git, GitHub, FastAPI, Docker, Large Language Model APIs or Local Models, and Machine Learning Development Tools.

Course Certification: Course Completion Certificate will be provided by Tech Learniversity.

Expected Learning Outcomes

Upon completion of the program, learners should be able to write Python programs, clean and analyze datasets using NumPy and Pandas, apply mathematical and statistical concepts, and develop predictive Machine Learning models using supervised and unsupervised learning algorithms.

Learners will develop practical skills in Python Programming, Data Preprocessing, Exploratory Data Analysis (EDA), Regression, Classification, Clustering, Feature Engineering, Model Evaluation, Hyperparameter Optimization, and Reproducible Machine Learning Pipelines.

The program also introduces Deep Learning, Artificial Neural Networks, Natural Language Processing (NLP), Computer Vision, Generative AI, Large Language Models (LLMs), Prompt Engineering, Retrieval-Augmented Generation (RAG), and Responsible AI Development.

Learners will gain practical experience developing introductory neural network models, text classification applications, image classification systems, LLM-powered workflows, document-based AI assistants, and prediction applications using Python, TensorFlow/Keras, Hugging Face, and FastAPI.

Through guided coding exercises, Machine Learning experiments, practical assessments, and a 22-hour integrated capstone project, learners will develop hands-on experience in model training, evaluation, error analysis, API integration, application prototyping, technical documentation, and project presentation.

The program also introduces MLOps Fundamentals, Model Versioning, Experiment Tracking, Deployment Concepts, Docker, Git/GitHub, AI Application Testing, and Model Monitoring, helping learners understand the practical development lifecycle of AI and Machine Learning solutions.

By completing the required assessments and final capstone, learners will develop a practical AI and Machine Learning project portfolio demonstrating their programming skills, analytical reasoning, predictive modeling capabilities, application development knowledge, and technical communication skills relevant to Junior Machine Learning Project Contributor, Entry-Level AI Application Developer, Generative AI Application Builder, and related junior technical roles.

Syllabus

Module 01. Introduction to Artificial Intelligence

- AI history and major branches; narrow AI versus broader AI aspirations
- Traditional rules-based programming versus learning from data
- Supervised, unsupervised and reinforcement learning overview
- AI applications across education, healthcare, finance, marketing and operations
- Generative AI, LLM capabilities and limitations; hallucination and verification
- Ethics: data privacy, bias, accountability, consent and human oversight

Hands-on labs and practical work

- Compare three AI use cases and identify inputs, outputs, data and limitations
- Try an AI assistant, verify factual claims and document risks

Module assessment / evidence

- One-page AI use-case canvas plus 10-question baseline diagnostic.

Module 02. Python Programming Fundamentals

- Python, Colab, Jupyter, environment basics and notebooks
- Variables, types, operators, I/O, conversion, conditionals and loops
- Strings, lists, tuples, sets, dictionaries and comprehensions
- Functions, arguments, return values, modules and imports
- File I/O, CSV and JSON, exceptions, debugging and simple logging
- Reusable functions, basic testing with assertions, style and documentation
- Introduction to Git, GitHub, repositories, commits and README files

Hands-on labs and practical work

- Build a student-records CLI/notebook with validation, summaries and CSV export
- Solve 15 progressive coding exercises and push a clean repository

Module assessment / evidence

- Timed Python practical (functions, files, validation) plus GitHub code submission.

Module 03. Mathematics and Statistics for AI

- Algebra, functions, percentages, ratios and logarithm intuition
- Descriptive statistics, variance, standard deviation and data distributions

- Probability, conditional probability and Bayes intuition
- Vectors, matrices, dimensions, dot products and transformations
- Correlation, covariance and the correlation-versus-causation distinction
- Slope, derivatives and gradient descent intuition; loss functions

Hands-on labs and practical work

- Compute metrics and matrix operations using NumPy
- Visualize a linear regression loss curve and a few gradient updates

Module assessment / evidence

- Worked statistics worksheet and a short annotated numerical notebook.

Module 04. Data Analysis with Python

- NumPy arrays, vectorization and broadcasting
- Pandas Series/DataFrames, filtering, grouping, joins and pivot tables
- Reading CSV, Excel and JSON; missing values, duplicates and outliers
- Data profiling, exploratory data analysis and visualization with Matplotlib/Seaborn
- SQL basics: SELECT, WHERE, GROUP BY, JOIN and querying SQLite
- Reproducible notebooks, effective charts and communicating findingsHands-on labs and practical work
- Analyze a sales dataset including trends, seasonality and customer groups
- Write five SQL queries and reconcile results with Pandas

Module assessment / evidence

- EDA report with data dictionary, reproducible notebook and five interpreted charts.

Module 05. Machine Learning Fundamentals

- ML problem definition, features, labels and prediction targets
- Supervised, unsupervised and reinforcement learning use cases
- Classification, regression, clustering and recommendation overview
- Train/validation/test splits; stratification and temporal splits
- scikit-learn estimator API, fitting, prediction and baseline models
- Ethical data sourcing, leakage risks and reproducibility

Hands-on labs and practical work

- Create baseline regression and classification models using scikit-learn
- Draft a problem statement and specify success metrics for a public dataset

Module assessment / evidence

- Baseline notebook and short ML workflow quiz.

Module 06. Supervised Machine Learning

- Linear, multiple and polynomial regression with assumptions
- Logistic regression and probabilistic outputs
- Decision trees, random forests and gradient boosting
- K-nearest neighbors, naive Bayes and support vector machines
- Training objectives, hyperparameters and interpretation
- Feature importance, simple error analysis and model comparison

Hands-on labs and practical work

- Predict housing prices with linear and tree-based methods
- Build churn or loan-risk classification with at least four models
- Compare a simple baseline and advanced model on consistent splits

Module assessment / evidence

- Supervised ML comparison report with metrics, plots, limitations and source code.

Module 07. Unsupervised Machine Learning

- Clustering use cases and distance metrics
- K-means, hierarchical clustering and DBSCAN
- Choosing parameters, silhouette score and cluster stability
- PCA, scaling and visualization of high-dimensional features
- Cluster labeling, limitations and fairness considerations

Hands-on labs and practical work

- Customer segmentation using behavioral variables
- Compare two clustering methods and interpret segment profiles

Module assessment / evidence

- Customer segmentation notebook and stakeholder-facing segment summary.

Module 08. Model Evaluation and Optimization

- Confusion matrix, precision, recall, F1, accuracy and class imbalance
- ROC-AUC, precision-recall curves, threshold decisions and calibration intuition

- MAE, MSE, RMSE, R-squared and residual diagnostics
- Overfitting, underfitting, bias-variance and learning curves
- Cross-validation, stratification and time-series validation
- Grid search, randomized search, model selection and test-set discipline
- Model slices, basic fairness and documentation

Hands-on labs and practical work

- Tune a classification pipeline while keeping a held-out test set untouched
- Perform error analysis and threshold selection for an imbalanced case

Module assessment / evidence

- Model-evaluation worksheet, tuning experiment table and test report.

Module 09. Feature Engineering and ML Pipelines

- Missing-value imputation, categorical encoding and scaling
- Date/time, text-length and domain feature creation
- Feature selection, transformations and imbalanced-data strategies
- scikit-learn Pipeline and ColumnTransformer
- Consistent train/inference preprocessing and serialization
- Pipeline testing, feature lineage and reproducibility

Hands-on labs and practical work

- Create end-to-end preprocessing for a tabular dataset
- Save, reload and test a model pipeline on new input records

Module assessment / evidence

- Reproducible pipeline repository with tests and documented inputs.

Module 10. Deep Learning and Neural Networks

- Perceptrons, dense networks, weights, biases, activation functions
- Forward propagation, losses, backpropagation intuition and optimizers
- Learning rates, epochs, batch sizes, dropout and regularization
- TensorFlow/Keras model construction, fitting and evaluation
- Validation curves, early stopping, initialization and hyperparameter experiments
- CPU/GPU notebooks, reproducibility and compute constraints

Hands-on labs and practical work

- Train a neural network for tabular or image data
- Experiment with architecture, learning rates and dropout; compare learning curves

Module assessment / evidence

- Neural network experiment log with model card and reproducible notebook.

Module 11. Natural Language Processing

- Text cleaning, tokenization, stemming, lemmatization and stop words
- Bag of Words, n-grams, TF-IDF and text classification
- Sentiment analysis, spam detection and evaluation
- Word embeddings, contextual embeddings and semantic similarity
- Transformer-based NLP introduction and responsible text processing

Hands-on labs and practical work

- Build spam or sentiment classification using TF-IDF plus logistic regression
- Compare predictions with an embedding-based or pretrained model baseline

Module assessment / evidence

- NLP classification notebook, error categories and documented limitations.

Module 12. Computer Vision

- Image dimensions, color channels, preprocessing and normalization
- Image augmentation, train/test contamination and dataset splits
- CNNs, convolution, pooling and feature maps
- Transfer learning using a pretrained vision backbone
- Image classification, confusion matrices and model inspection
- Object detection overview and ethical image-data use

Hands-on labs and practical work

- Create an image classifier with transfer learning
- Inspect incorrect predictions and test augmentations

Module assessment / evidence

- Vision demo with metrics, sample predictions, dataset statement and model card.

Module 13. Generative AI and Large Language Models

- Generative AI and LLM fundamentals, tokens, embeddings and context windows
- Transformer attention at a conceptual level; prompting and instruction hierarchy
- Prompt patterns, few-shot examples, output schemas and structured JSON
- API integration, model parameters, rate limits and token cost estimates
- Function/tool calling and agent workflow concepts; deterministic versus autonomous steps
- Evaluation sets, factuality, hallucinations, injection defense and content safety
- Privacy, sensitive data handling, licensing and human approval

Hands-on labs and practical work

- Build a structured information extractor with validation
- Design a tool-calling workflow with explicit permission boundaries
- Benchmark a small prompt set and document accuracy, latency and cost

Module assessment / evidence

- Small LLM application with evaluation dataset, safety notes and cost estimate.

Module 14. Retrieval-Augmented Generation (RAG)

- RAG architecture, versus fine-tuning, and knowledge-source decisions
- Document parsing, chunking, overlap and metadata management
- Embeddings, vector indexes, similarity and hybrid retrieval concepts
- Search filtering, top-k, reranking overview and context selection
- Grounded answer generation, citations, abstention and fallback behavior
- Retrieval recall/precision intuition, answer faithfulness and test sets
- Security: access control, untrusted documents and prompt injection

Hands-on labs and practical work

- Build a document question-answering assistant with source citations
- Test at least 20 queries covering answerable, unanswerable and adversarial cases

Module assessment / evidence

- RAG application demo, retrieval evaluation sheet and failure analysis.

Module 15. AI Application Development and Deployment

- FastAPI routes, schemas, validation and error responses
- Loading serialized models; inference workflow and API tests

- Streamlit or lightweight web front end for user interaction
- Container basics with Docker, configuration and environment variables
- Deployment to a suitable hosting environment; health checks and logging
- Authentication concepts, secrets management, input limits, error handling and cost awareness

Hands-on labs and practical work

- Build an inference API and simple UI for a project model
- Dockerize, deploy where feasible and document an alternative reproducible local demo

Module assessment / evidence

- Working application prototype, README, deployment instructions and smoke tests.

Module 16. MLOps and Production AI Fundamentals

- Git collaboration, branch and review workflow
- Data/model versioning, repeatable training and experiment logs
- Model registry concepts and release checkpoints
- CI/CD foundations and automated checks for ML projects
- Prediction monitoring, drift, performance degradation and retraining triggers
- Incident logs, rollback plans, privacy and operational responsibilities

Hands-on labs and practical work

- Track two model versions and compare experiment results
- Create a simple monitoring dashboard or monitoring specification

Module assessment / evidence

- MLOps checklist, versioned project release and monitoring plan.

Module 17. Capstone Project

- Project proposal, stakeholder problem, feasibility and success metrics
- Data sourcing, consent/licensing, EDA and risk review
- Preprocessing, training or RAG pipeline design and baselines
- Evaluation against held-out or curated test sets; error and failure analysis
- Usable notebook, UI and/or API; deployment or reproducible run instructions
- Documentation, model card, demonstration, presentation and peer feedback

Hands-on labs and practical work

- Build a portfolio-quality project under instructor checkpoints
- Present a working demo and defend evaluation methods and limitations

Module assessment / evidence

- Capstone repository, working demo, report, slides and oral viva scored by rubric.