MACHINE LEARNING PROFESSIONAL PROGRAM – 2025 EDITION

"From algorithms to intelligent systems — build the future with ML."

Market Demand Note

Machine Learning is at the core of AI transformation. Companies across sectors — from finance to healthcare — are seeking ML engineers to automate decision-making, improve efficiency, and personalize user experiences. This program focuses on practical skills with real-world datasets.

Duration: 12 Weeks | Mode: Online/Offline

📘 Machine Learning Professional Program — 2025 Edition

Key Tools & Technologies:
Python, Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, TensorFlow, PyTorch, OpenCV, Git, Jupyter Notebook, Kaggle Datasets.

Learning Objectives

By the end of this program, learners will be able to:

1. Understand and implement core ML algorithms from scratch and using libraries.
2. Preprocess and engineer features for optimal model performance.
3. Apply supervised, unsupervised, and ensemble learning techniques.
4. Deploy ML models into production environments.
5. Solve real-world problems with end-to-end ML workflows.

Table of Contents

Detailed Content

Week 1: ML Foundations & Environment Setup

- Introduction to ML: Real-world applications across industries (healthcare, finance, e-commerce, IoT).
- ML lifecycle: From problem definition to deployment and monitoring.
- Python, Jupyter, and essential libraries: Install and configure Anaconda, JupyterLab, pandas, numpy, scikit-learn.
- Types of data: Structured, unstructured, tabular, image, text.
- Hands-on: Set up environment, load and explore sample datasets, run first ML workflow.
- Current trend: Cloud notebooks (Google Colab, Kaggle Kernels) are increasingly used for collaboration and scalability.

Week 2: Data Preparation

- Missing data: Imputation (mean, median, mode), deletion, advanced methods (KNN, iterative).
- Categorical encoding: One-hot, label, target, embeddings (for high-cardinality).
- Feature scaling: Standardization, normalization, robust scaling.
- Initial feature engineering: Creating new features, binning, interaction terms.
- Hands-on: Clean a real-world messy dataset (e.g., sales, customer, IoT sensor data).
- Industry relevance: Data quality is the #1 reason ML projects fail—employers prioritize candidates who can handle real, messy data.

Week 3: Supervised Learning: Regression Models

- Linear regression: Theory, assumptions, interpretation.
- Regularization: Ridge, Lasso regression—when and why to use.
- Evaluation metrics: MSE, RMSE, R^2, MAE.
- Practical modeling: Train, validate, and interpret regression models with scikit-learn.
- Hands-on: Predict house prices or sales using real datasets.
- Trend: Explainability and model interpretability (SHAP, LIME) are increasingly required in production ML.

Week 4: Supervised Learning: Classification Models

- Logistic regression: Binary and multiclass, interpretation, decision boundaries.
- Decision trees and random forests: How they work, pros/cons, feature importance.
- Evaluation metrics: Accuracy, precision, recall, F1, ROC-AUC, confusion matrix.
- Overfitting/underfitting: Detection and mitigation strategies.
- Hands-on: Build a fraud detection or customer churn classifier.
- Industry demand: Classification is foundational for analytics, risk, and automation roles.

Week 5: Ensemble Methods

- Bagging: Random Forests—concept, implementation, tuning.
- Boosting: XGBoost, LightGBM—gradient boosting, categorical feature handling, GPU support.
- Stacking: Combining multiple models for improved performance.
- Hyperparameter tuning: Practical grid/random search with scikit-learn.
- Hands-on: Compete on a Kaggle dataset using ensembles.
- Current tools: XGBoost, LightGBM, CatBoost are industry standards for structured data.

Week 6: Unsupervised Learning

- Clustering: K-Means (elbow method, silhouette), DBSCAN (density-based), hierarchical.
- Dimensionality reduction: PCA (explained variance), t-SNE (visualization), UMAP.
- Applications: Customer segmentation, anomaly detection, feature extraction.
- Hands-on: Segment customers or reduce dimensions of a high-dimensional dataset.
- Trend: Unsupervised techniques are critical for exploratory analysis and feature engineering.

Week 7: Model Selection & Optimization

- Cross-validation: k-fold, stratified, time series splits.

- Hyperparameter optimization: Grid search, random search, Bayesian optimization.
- Bias-variance tradeoff: Diagnosing and fixing overfitting/underfitting.
- Hands-on: Optimize a model pipeline end-to-end.
- Industry practice: Automated ML (AutoML) tools (Auto-sklearn, TPOT) are gaining traction for rapid prototyping.

Week 8: Deep Learning

- Neural network fundamentals: Perceptrons, activation functions, backpropagation.
- Building models: TensorFlow/Keras APIs, training loops, callbacks.
- Visualization: Training curves, metrics, embedding visualization.
- Hands-on: Train a neural net on tabular or image data.
- Trend: Deep learning is essential for NLP, vision, and advanced analytics roles.

Week 9: Computer Vision Basics

- Image preprocessing: Resizing, normalization, augmentation.
- Feature extraction: OpenCV for edges, corners, histograms.
- CNNs: Convolution, pooling, transfer learning (pre-trained models).
- Hands-on: Build an image classifier (e.g., medical imaging, product recognition).
- Industry application: Computer vision is booming in healthcare, retail, autonomous systems.

Week 10: Model Deployment

- Model serialization: joblib, pickle, ONNX.
- API development: Flask, FastAPI for model serving.
- Scaling and monitoring: Containerization (Docker), cloud deployment (AWS SageMaker, GCP AI Platform, Azure ML).
- Hands-on: Deploy a model as a REST API, containerize, and test.
- Trend: MLOps (CI/CD, monitoring, retraining) is now a core expectation for ML roles.

Week 11: Real-World Project Work

- Dataset selection: Kaggle, UCI, government open data.

- Full ML pipeline: Data prep, feature engineering, modeling, evaluation, deployment.
- Teamwork: Collaborate using GitHub, document process, review code.
- Hands-on: Deliver a complete ML solution on a real-world problem.
- Portfolio value: End-to-end projects are the top differentiator in job interviews.

Week 12: Capstone Project & Presentation

- Project delivery: Code, README, visualizations, deployment (if applicable).
- Presentation: Explain approach, results, business impact.
- Peer review: Give and receive constructive feedback.
- Assessment: Technical quality, documentation, automation, storytelling.
- Capstone ideas: House price prediction, fraud detection, medical image classification—all using real, messy data.

Current Tools & Industry Trends

- Python: pandas, numpy, scikit-learn, XGBoost, LightGBM, TensorFlow, Keras, OpenCV
- Cloud: AWS SageMaker, GCP AI Platform, Azure ML, Colab, Kaggle
- Deployment: Flask, FastAPI, Docker, Kubernetes
- Version Control: GitHub, GitLab
- Automation: Airflow, MLflow, DVC
- Ethics: Responsible AI, fairness, bias detection, explainability (now part of global curricula)

Capstone Project Examples

- House Price Prediction: Ingest, clean, and model real estate data; deploy as API.
- Fraud Detection: Build a classifier for transaction data, optimize for precision/recall.
- Medical Image Classification: Preprocess scans, train a CNN, interpret results.

Key Differentiators for Job Seekers

- End-to-end project experience (data to deployment).
- Clean, modular, documented code (GitHub portfolio).
- Exposure to MLOps and cloud tools (deployment, automation, monitoring).
- Storytelling and presentation skills (clear communication of insights).
- Ethical AI awareness (bias, fairness, explainability).