PRAXC INFINITE

PRACTICAL MACHINE LEARNING WITH PYTHON

An 8-Week Applied Learning Program | Beginner to Intermediate

COURSE SYLLABUS

Duration	8 weeks	**Suggested hours**	60 hours
Level	Beginner to Intermediate	**Delivery**	Live online / instructor-led / practical labs
Prerequisites	Basic Python recommended; refresher included	**Environment**	Jupyter Notebook / Google Colab

Why Learn Machine Learning with Python?

Machine Learning helps turn data into useful predictions and insights—for example, estimating prices, identifying patterns, grouping customers, and supporting data-driven decisions. Python is widely used for Machine Learning because its ecosystem includes accessible tools for data preparation, visualization, model training, and evaluation. This course is designed for learners who want to move beyond theory and understand how to build and assess practical ML models step by step.

Course Description

The Practical Machine Learning with Python program takes learners from core concepts to applied modelling over eight weeks. Participants will use NumPy and Pandas to prepare data, Matplotlib and Seaborn to explore patterns, and Scikit-learn to build regression, classification, and clustering solutions. Guided exercises, real-world-style datasets, and a capstone project help learners practise the full workflow—from defining a problem and preparing data to training models, evaluating results, and presenting findings. The program is suitable for beginners with basic Python familiarity and learners preparing for further study in Machine Learning, Artificial Intelligence, or Data Science.

Learning Outcomes

- Explain key ML concepts and distinguish common learning approaches.
- Use Python libraries to clean, transform, and visualize datasets.
- Perform Exploratory Data Analysis and summarize findings.
- Prepare model-ready data while avoiding common data leakage.
- Train regression and classification models using Scikit-learn.
- Evaluate models using suitable metrics and compare performance.
- Apply introductory clustering, cross-validation, and model tuning.
- Complete and present a practical ML project, including limitations.

Tools & Technologies

Programming: Python | Notebook: Jupyter Notebook / Google Colab

Data handling: NumPy, Pandas | Visualization: Matplotlib, Seaborn

Machine Learning: Scikit-learn

8-Week Curriculum

Week 1 — Machine Learning Foundations & Python Workflow

Module Description: Understand what Machine Learning is, why it is used, and how an ML project works from problem definition to evaluation.

Topics Covered

- AI, Machine Learning and Deep Learning: differences and relationship
- Traditional programming versus learning from data
- ML applications, strengths and limitations
- Supervised, unsupervised and reinforcement learning overview
- ML lifecycle, features, targets and datasets
- Jupyter Notebook / Google Colab setup
- Python refresher: variables, collections, functions, loops and imports
- Introduction to NumPy, Pandas and Scikit-learn

Hands-on Activity: Load a sample dataset, identify its features and target, and define a prediction task.

Learning Outcome: Explain ML fundamentals and outline a basic workflow.

Week 2 — Python for Data Handling with NumPy & Pandas

Module Description: Develop practical data-handling skills for working with structured datasets.

Topics Covered

- NumPy arrays, indexing, slicing and vectorized operations
- Pandas Series and DataFrames
- Read CSV and Excel files
- Inspect shape, columns, types and summary statistics
- Select, filter, sort and group records
- Handle missing values and duplicates
- Correct data types and transform columns
- Combine datasets and create derived features
- Reproducible notebook workflows

Hands-on Activity: Explore and clean a business or student-performance dataset.

Learning Outcome: Load, inspect, clean and transform tabular data in Python.

Week 3 — Exploratory Data Analysis & Visualization

Module Description: Learn to identify patterns, distributions and relationships before building a model.

Topics Covered

- Data quality checks and descriptive statistics
- Numerical and categorical variables
- Univariate and bivariate analysis
- Matplotlib charts: line, bar, histogram and scatter
- Seaborn: box plots, count plots, pair plots and heatmaps
- Distributions, outliers and correlation
- Avoiding misleading visualizations
- Exploratory Data Analysis workflow
- Using findings to guide model preparation

Hands-on Activity: Create an EDA notebook with charts and a concise findings summary.

Learning Outcome: Interpret data patterns and communicate useful observations.

Week 4 — Data Preparation & Feature Engineering

Module Description: Prepare reliable model-ready data and understand how preprocessing affects performance.

Topics Covered

- Defining target variables and selecting features
- Train/test split and data leakage

- Handling missing values and outliers
- Encoding categorical variables
- Feature scaling and standardization
- Feature selection and simple feature engineering
- Scikit-learn preprocessing pipelines
- Introductory class imbalance
- Common preprocessing mistakes

Hands-on Activity: Prepare a mixed-type dataset for modelling with a repeatable workflow.

Learning Outcome: Prepare data for modelling while avoiding test-data leakage.

Week 5 — Supervised Learning: Regression

Module Description: Use regression algorithms to predict numerical values and measure prediction errors.

Topics Covered

- Supervised learning and baseline models
- Simple and multiple Linear Regression
- Understanding coefficients and predictions
- Introductory regression assumptions
- Decision Tree Regression
- Random Forest Regression
- Train models with Scikit-learn
- MAE, MSE, RMSE and R^2
- Residuals and prediction errors
- Compare model performance

Hands-on Activity: Build a house-price, sales or student-score prediction model.

Learning Outcome: Train and evaluate regression models and interpret their errors.

Week 6 — Supervised Learning: Classification

Module Description: Build models that predict categories and evaluate their performance appropriately.

Topics Covered

- Classification tasks and class labels
- Logistic Regression
- K-Nearest Neighbours (KNN)
- Decision Tree and Random Forest classifiers
- Prediction probabilities and thresholds
- Confusion matrix
- Accuracy, precision, recall and F1-score
- Class imbalance and limitations of accuracy
- Choosing metrics for the use case
- Compare classification models

Hands-on Activity: Build a classification model and interpret its confusion matrix and metrics.

Learning Outcome: Train classifiers and select relevant evaluation measures.

Week 7 — Unsupervised Learning, Validation & Model Improvement

Module Description: Explore clustering and learn how to assess generalization and compare models.

Topics Covered

- Unsupervised learning and clustering applications
- K-Means and selecting cluster count
- Scaling data for distance-based algorithms
- Interpreting and visualizing clusters
- Overfitting, underfitting and generalization
- Training, validation and test sets
- Cross-validation fundamentals
- Bias-variance intuition
- Introductory hyperparameter tuning

- GridSearchCV / RandomizedSearchCV
- Fair model comparison

Hands-on Activity: Perform customer or learner segmentation and compare model configurations.

Learning Outcome: Explain clustering, recognize overfitting and compare models systematically.

Week 8 — End-to-End Machine Learning Project

Module Description: Apply the full workflow in a guided project and communicate results clearly.

Topics Covered

- Project planning and dataset selection
- Define the problem and success criteria
- Load, inspect and clean data
- Exploratory analysis and visualization
- Feature preparation and train/test split
- Select and train an algorithm
- Evaluate and compare models
- Interpret results, limitations and possible bias
- Document code and communicate findings
- Save a trained model: introduction
- Deployment concepts overview
- Project presentation and feedback

Hands-on Activity: Complete a guided project such as price prediction, churn classification or customer segmentation.

Learning Outcome: Demonstrate an end-to-end ML workflow and explain results and limitations.

Training Methodology

- Instructor-led explanations with practical demonstrations.
- Guided coding in Jupyter Notebook or Google Colab.
- Weekly exercises using sample and realistic datasets.
- Short reviews to reinforce concepts and technical vocabulary.
- Capstone project covering data preparation, modelling, evaluation, and communication.

Suggested Assessment Framework

Assessment Component	Suggested Weight
Weekly coding exercises and concept checks	20%
Data preparation and exploratory analysis	20%
Model training and evaluation tasks	25%
Final project, documentation, and presentation	35%

Who Should Attend?

College students, graduates, Python learners, aspiring Machine Learning practitioners, and professionals who want hands-on experience with predictive modelling. Basic Python knowledge is recommended; the first week includes a refresher.

Suggested Capstone Projects

- House-price or sales prediction using regression
- Customer churn or student-performance classification
- Customer segmentation using clustering

PRAXC INFINITE | AI & TECHNOLOGY EDUCATION
Learn Today, Lead Tomorrow