Suggested Teaching Guidelines for

Practical Machine Learning
PG-DBDA August 2025

Duration: 60 hours Theory and 80 hours Lab

Objective: Practicing Machine Learning Algorithms

Prerequisites: Good knowledge of Python Programming and Statistics

Evaluation method:	Theory exam– 40% weightage Lab exam – 40% weightage Internal exam– 20% weightage

<u>List of Books / Other training material</u>

Textbook:
Machine Learning, Saikat Dutt / Pearson

Reference Book:
Machine Learning using Python , Manaranjan Pradhan , U Dinesh Kumar, Wiley

Note: Each session having 2 Hours of Theory & 2 Hours of Lab unless mentioned otherwise.

Session 1:

- Fundamentals of information theory
- Introduction to machine learning
- Algorithm types of Machine learning
- Probably Approximately Correct (PAC) Learning
- Uses of Machine learning
- Evaluating ML techniques

Session 2:

- Bias complexity trade off
- Vapnik-Chervonenkis (VC) Dimension
- Non-uniform learnability (Structural risk minimization, Occam's Razor and No Free Lunch Theorem)
- Regularization and Stability

Suggested Teaching Guidelines for

Practical Machine Learning

PG-DBDA August 2025

Session 3:

- Model Selection and Validation
- Introduction to Scikit Learn
- Performing ML using Scikit Learn

Assignments:

- Explore scikit learn Library.

- Explore Datasets Online (can refer Kaggle, UCI ML, etc.)
    - Load dataset in google colab.
    - Print first five values and last five values in dataset.
    - check correlation between fields present in dataset

Session 4 & 5:

- Clustering
- Hierarchical Clustering & K means
- Distance Measure and Data Preparation – Scaling & Weighting
- Evaluation and Profiling of Clusters
- Hierarchical Clustering
- Clustering Case Study
- Principal Component analysis: PCA, Kernel PCA
- Random Projections

Assignments:

Download “mall_customers.csv” dataset from Kaggle.

(a) Form n no. of clusters according to your observation.
(b) Get wss value for each cluster.
(c) find best K value

Session 6 & 7: (4T + 6L)

- Evaluation metrics: Accuracy, F1-score, ROC AUC, Log Loss
- Evaluation metrics: MAE, RMSE, R2 score
- Decision Trees
- Classification and Regression Trees
- Random forest, Gradient boosting Machines, Model Stacking
- CAT Boost & Light GBM
- XG Boost

Suggested Teaching Guidelines for

Practical Machine Learning

PG-DBDA August 2025

Assignments:

- Implement Random Forest, SVM, Logistic regression classification algorithm
- Check for classification report, f1 score for all three algorithms.

Session 8 & 9:

- Bayesian analysis and Naïve bayes classifier
- Assigning probabilities and calculating results
- Linear Discriminant Analysis
- K-Nearest Neighbors Algorithm

Assignments:

- Implement K-Nearest Neighbors Algorithm

Session 10 & 11: (4T + 6L)

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- Elastic Net Regression

Assignments:

- Download Dataset, perform linear, Ridge, Lasso, Polynomial regression and check for MAE, MSE, RMSE, F1 score and explain with conclusion.

Session 12: (2T + 4L)

- Support Vector Machines
- Basic classification principle of SVM
- Linear and Nonlinear classification (Polynomial and Radial)

Assignments:

- Download Air Quality Dataset from Kaggle Predict Air Quality Index using Linear regression and classify it into five categories using SVM (i.e. Very good, good, moderate, poor, worst)

Suggested Teaching Guidelines for

Practical Machine Learning

PG-DBDA August 2025

Session 13 & 14:

- Moving average, Exponential Smoothing, Holt's Trend Methods, Holt-Winters'
- Methods for seasonality
- Autocorrelation (ACF & PACF), Auto-regression, Auto-regressive Models, Moving Average Models
- ARMA &ARIMA

Assignments:

- What is Auto correlation, explain its purpose Also download one data set and calculate Auto correlation.
- Explain ARMA and ARIMA model, what is purpose of this models in time series and Explain difference between them.

Session 15 & 16: (4T + 6L)

- Recommendation Systems
  - Data Collection & Storage, Data Filtering
  - Collaborative Filtering
  - Factorization Methods
  - Evaluation Metrics: Recall, Precision, RMSE, Mean Reciprocal Rank, MAP at K, NDCG

Session 17, 18 & 19:

- Introduction to Deep Learning
- Introduction to Tensor flow and Keras

Assignments:

Explore Tensor Flow and Keras Libraries

Session 18 & 19: (4T + 6L)

- Introduction to Auto-encoders
- Neural Network and its applications
- Single layer neural Network
- Activation Functions: Sigmoid, Hyperbolic Tangent, ReLu
- Overview of Back propagation of errors

Assignments:

- Implement Different Activation functions on datasets in Jupyter Notebook.

Session 20:

Deep Learning Essentials

- Early Stopping for Preventing Overfitting
  - Dropout
  - L1 and L2 Regularization
- Update of weights with single training set element

Session 21 & 22: (4T + 6L)

- Batch Training, Mini-batch Training, Stochastic Gradient Descent
- Training Methods for Neural Network (High-Level Overviews only) / Opimizers
  - Classic Back propagation
  - Momentum Back propagation
  - ADAM

Assignments:

- Implement Different optimizers
- Implement Gradient Problems

Session 23 & 24: (4T + 6L)

Convolutional Neural Network using PyTorch

- Introduction to PyTorch Framework
- Pytorch vs Tensor flow
- Convolutional Concept
- Transfer Learning and Inception Network as one of its examples
- Data Augmentation
- Object Detection
- YOLO Algorithm (High-Level Overview)

Assignments:

- Install PyTorch. Explore the documentation of PyTorch Library.
- Implement YOLO Algorithm.

Session 25 & 26: (4T + 6L)

Recurrent Neural Network (RNN) using Pytorch

- RNN Concept
- Types of RNNs
- Vanishing gradients with RNNs

- Gated Recurrent Unit (GRU) - (High-Level Overview only)
- Long Short-Term Memory (LSTM) - (High-Level Overview only)

Assignments:

- Implement RNN using PyTorch
- Implement LSTM and GRU

Case Studies:

- Time series example with LSTM

Session 27: (2T + 4L)

Generative AI: Introduction to Transformers

- Transformers and their importance
- A brief history of NLP and the rise of transformers (including the Attention Timeline)
- Transformers
  - Introduction to BERT (Bidirectional Encoder Representations from Transformers):
  - Pre-training objectives of BERT and its impact on NLP tasks
  - Fine-tuning BERT for various NLP applications (e.g., sentiment analysis, question answering)

  - Understanding and Handling Text Data:
    - Text pre-processing techniques (e.g., tokenization, stemming, lemmatization)
    - Representing text data for machine learning models
      - Hands-on activity: Pre-processing text data and fine-tuning a pre-trained model for a specific task
  - Applications of transformers across various domains (e.g., computer vision, speech recognition)

Suggested Teaching Guidelines for

Practical Machine Learning

PG-DBDA August 2025

Assignment:

- Hands-on activity: Exploring a pre-trained transformer model (e.g., using Google AI's Colab

Session 28

- Core Concepts of AI architectures
- Understanding the Encoder-Decoder Architecture:
  - The role of encoders in processing input sequences
  - The role of decoders in generating output sequences
  - Encoder-decoder for tasks like machine translation and text summarization

- Attention Mechanisms:
  - Demystifying attention and its variants (e.g., self-attention, masked attention)
  - How attention helps models "focus" on relevant parts of the input
- Hands-on activity: Implementing a simple attention mechanism

Session 29 & 30: (4T + 6L)

- Advanced Concepts & Applications
  - Introduction to Large Language Models (LLMs)
    - Capabilities and limitations of LLMs
    - Responsible development and use of LLMs
  - Reward Models and Alignment Strategies
  - Ensuring alignment between LLM goals and human values
  - Techniques for mitigating bias and promoting safety
- Practical Case Studies:
    - Exploring real-world applications of transformers and LLMs (e.g., chatbots, text generation, code completion)
- Deployment Considerations for LLMs:
    - Infrastructure requirements and challenges
    - Strategies for efficient deployment of LLMs in production environments

Assignments:

- Exploring an LLM API or building a simple application using a pre-trained model