4. Machine Learning Development Life Cycle (MLDLC/MLDC)

Wednesday, 21 January 2026 9:46 AM

In Software Development We have SDLC - Software Engineering.
Same in Machine Learning - MLDLC

MLDLC- It's guideline when you create a Machine Learning based software.

Idea is very simple you need to create a software and you need to integrate machine learning models into it end to end till deployment.

There might be different phase it's depend upon author.

1. Frame the Problem
2. Gathering the data
3. Data Preprocessing
4. Exploratory Data Analysis
5. Feature Engineering
6. Model Training
7. Model Deployment
8. Testing
9. Optimize

Machine Learning Development Cycle (Step-by-Step)

> The **names and order of phases may vary by author**, but the **core idea remains the same**.

1. Frame the Problem

What are we trying to solve?

- Define the **problem statement**
- Decide the **type of ML problem**
  - Classification / Regression / Clustering

Example:

- "Predict whether a student will **PASS or FAIL** based on attendance and marks."
- Type: **Classification**

Key Questions:

- What is input (features)?
- What is output (label)?
- How will success be measured?

2. Gathering the Data

Collect data needed to solve the problem

Sources:

- CSV files
- Databases
- APIs
- Surveys
- Public datasets (Kaggle, UCI)

Example:

Student dataset:

Attendance | Internal Marks | Assignment Score | Result

Attendance | Internal Marks | Assignment Score | Result
Without data → **No Machine Learning**

3. Data Preprocessing

Clean and prepare data for ML
Common tasks:

- Handle missing values
- Remove duplicates
- Encode categorical data
- Normalize/scale values

Example:

- Gender: Male/Female → 0/1
- Missing attendance → replace with average

Why important?

ML models work on **numbers**, not raw messy data.

4. Exploratory Data Analysis (EDA)

Understand data before modeling
Activities:

- Statistics (mean, min, max)
- Graphs (histogram, scatter plot)
- Finding patterns & outliers

Example:

- Students with attendance < 60% mostly fail
- Higher internal marks → higher pass rate

Tools:

- Pandas, Matplotlib, Seaborn

5. Feature Engineering

Create better input features for the model
Includes:

- Selecting important columns
- Creating new features
- Removing irrelevant features

Example:

- Create a new feature:

Total_Score = Internal + Assignment

Better features → Better model performance

6. Model Training

Teach the machine using data
Steps:

- Split data → Train & Test
- Choose algorithm
- Fit model

Examples:

- Linear Regression → Salary prediction
- Logistic Regression → Pass/Fail
- Decision Tree → Loan approval

Simple explanation to students:

"Model learns patterns from past data."

7. Model Deployment

Make the model usable in real life
Ways to deploy:

- Web app
- Mobile app
- API
- Cloud server

Example:

- A web app where teacher enters marks → model predicts PASS/FAIL
  Without deployment, ML is just a lab experiment.

8. Testing

Check how well the model works

Metrics:

- Accuracy
- Precision, Recall
- RMSE (Regression)

Example:

- Model accuracy = 87%
- Check predictions on **new unseen data**

Goal:

Ensure model generalizes, not memorizes.

9. Optimize (Model Improvement)

Improve performance continuously

Optimization methods:

- Hyperparameter tuning
- Try different algorithms
- More data
- Better features

Example:

- Change max_depth in Decision Tree
- Replace Logistic Regression with Random Forest
  ML is an **iterative process**, not one-time work.

Problem → Data → Clean → Understand → Features → Train → Deploy → Test → Improve