AIML CSET-301

-Neeraj Baghel

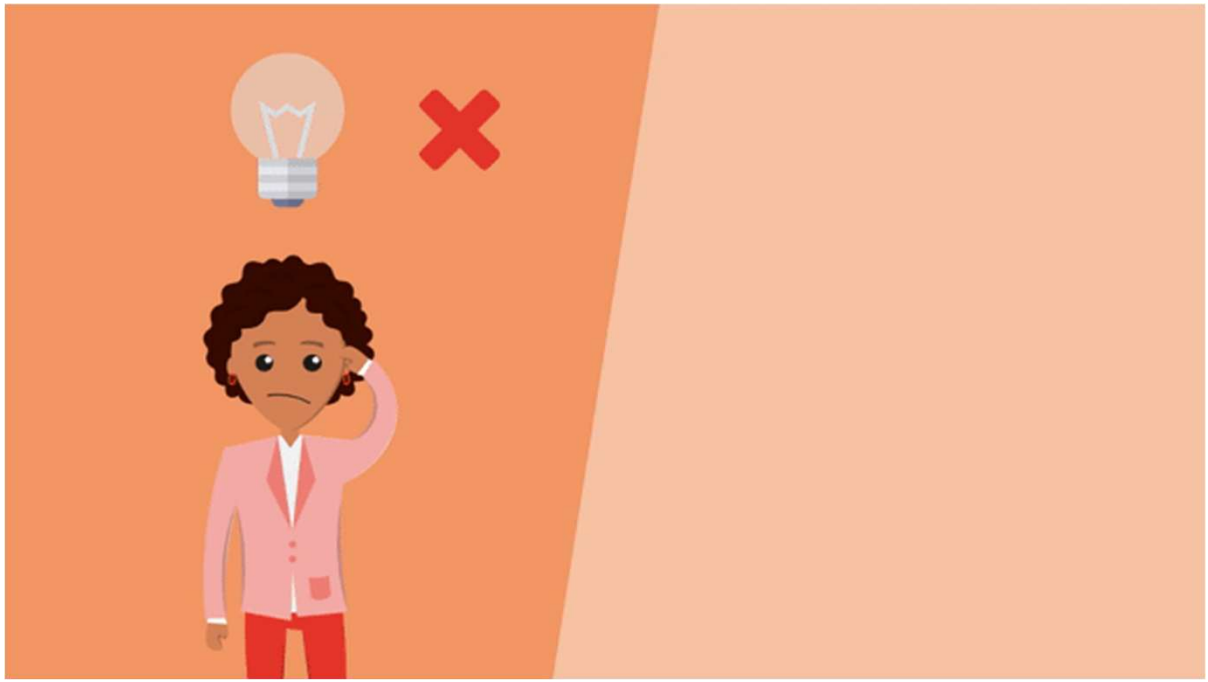
Generalized AIML Life cycle

1. Problem Identification
2. Gathering Data
3. Data Pre-processing
4. Exploratory Data Analysis
5. Model Selection
6. Train Model
7. Hyperparameter Tunning
8. Test Model
9. Deployment

AIML
Life cycle

1. Problem Identification

Machine learning project typically begins with the problem definition.



AIML
Life cycle

2. Gathering Data:

This step includes the below tasks:

- Identify various data sources
- Collect data
- Integrate the data obtained from different sources

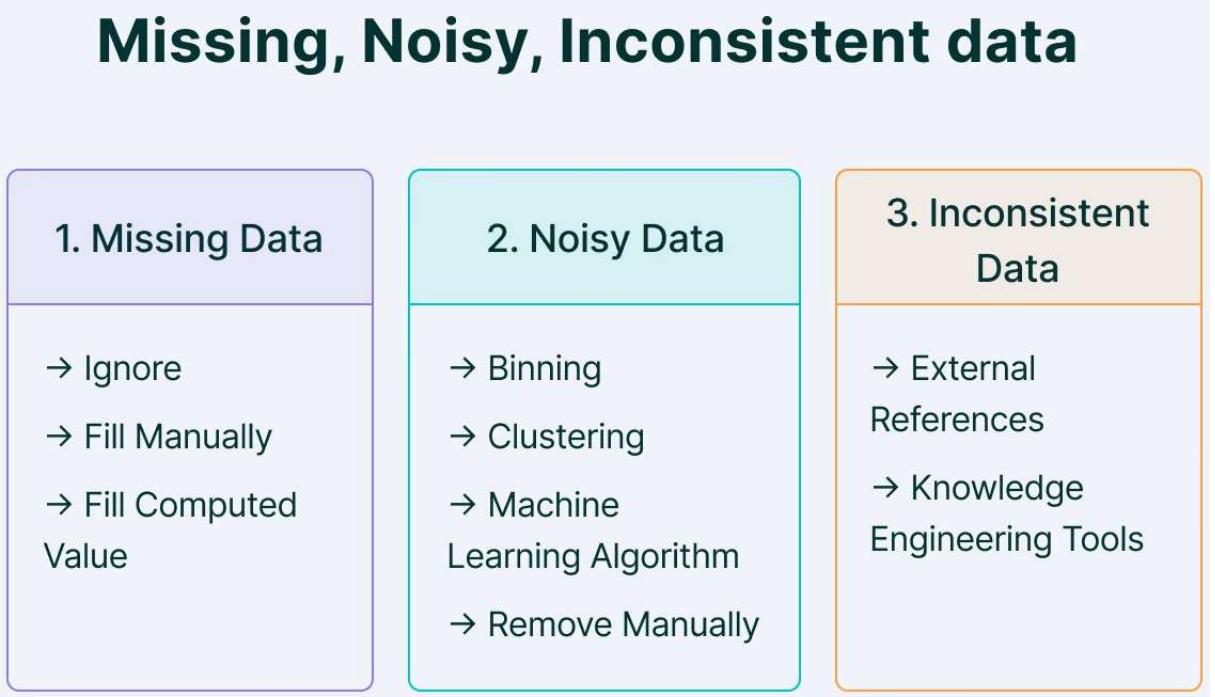
AIML
Life cycle

3. Data Pre-processing

Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

In real-world applications, collected data may have various issues, including:

- **Missing Values**
- **Duplicate data**
- **Invalid data**
- **Noise**



Focuses on cleaning and transforming data so that it can be used by machine learning algorithms. It is more about **data quality**.

AIML Life cycle

4. Exploratory Data Analysis:

- **Summary Statistics**:
Calculating and interpreting basic statistics (e.g., mean, median, mode, standard deviation) to get an overview of the data distribution.
- **Visualizations**:
Using plots (e.g., histograms, box plots, scatter plots, heatmaps) to visualize data distributions and relationships between variables.
- **Correlation Analysis**:
Identifying relationships between numerical variables (e.g., using correlation matrices).
- **Identifying Patterns & Anomalies**:
Finding interesting trends, clusters, or outliers in the data.
- **Forming Hypotheses**:
Based on visualizations and patterns, creating hypotheses for further testing and modeling.

Focuses on understanding and interpreting the data. It is more about **data understanding and insight generation**.

AIML Life cycle

5. Model Selection

Here we select the machine learning techniques such as **Classification**, **Regression and Cluster analysis**, etc. then build the model using prepared data, and evaluate the model.

AIML Life cycle

6. Train Model:

In this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms.

Training a model is required so that it can understand the various patterns, rules, and, features.

AIML
Life cycle

7. Hyperparameter Tunning

It is the process of finding the best set of hyperparameters for a machine learning model to maximize its performance on a specific task. Hyperparameters can include values such as:

- **Learning Rate:** Determines how quickly the model adapts to the problem.
- **Number of Layers/Neurons:** Defines the architecture of neural networks.
- **Regularization Parameters:** Controls overfitting (e.g., L1/L2 regularization).
- **Batch Size:** Number of samples used in each training iteration.
- **Number of Trees** (for ensemble methods like Random Forests).
- **Kernel Type** (in Support Vector Machines).

How to Implement Hyperparameter Tuning

1.Define a Search Space: Choose the range of possible values for each hyperparameter.

2.Choose a Tuning Technique: Select a method like grid search or random search.

3.Evaluate Model Performance:

1. Use cross-validation to assess the model for each combination.
2. Choose metrics like accuracy, precision, or F1-score.

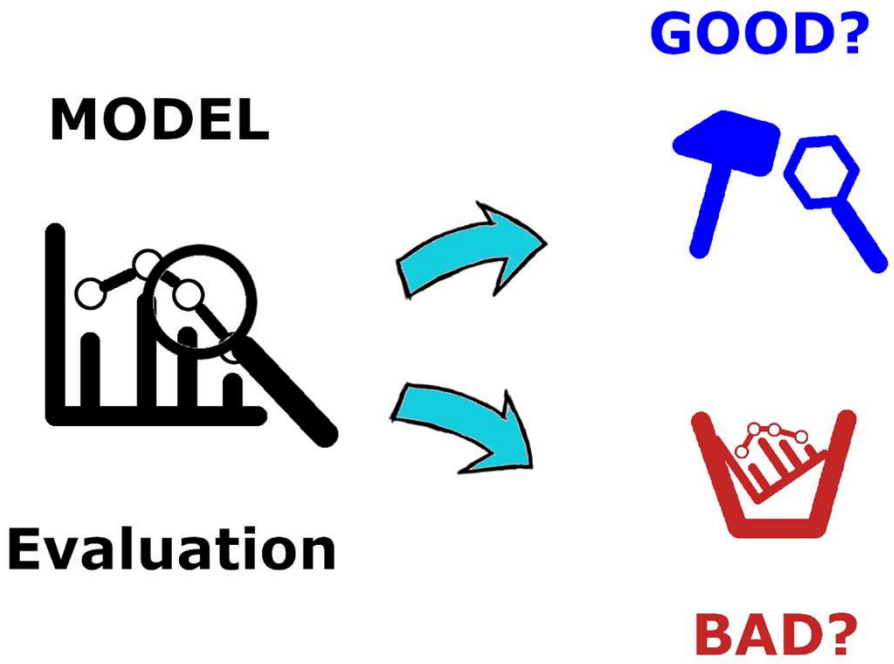
AIML
Life cycle

8. Test Model:

Once our machine learning model has been trained on a given dataset, then we test the model.

In this step, we check for the accuracy of our model by providing a test dataset to it.

Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.



AIML
Life cycle

9. Deployment:

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.

Key Steps in Deployment

1.Prepare the Model for Deployment:

1. Optimize the model for performance (reduce size, remove unnecessary components).
2. Convert the model to a format compatible with the deployment environment (e.g., TensorFlow Lite, ONNX, or PMML).

2.Select Deployment Environment:

1. **Cloud**: AWS, Azure, Google Cloud, etc.
2. **Edge Devices**: Smartphones, IoT devices, etc.
3. **On-Premise**: For systems requiring higher control or security.

3.Model Integration:

1. Embed the model into an application, API, or microservice for easy access.
2. Example: Using REST APIs to expose the model for use by external systems.

4.Set Up Monitoring and Feedback:

1. Implement monitoring systems to track model performance in real-time.
2. Capture user feedback and data for future improvements.

5.Scalability and Reliability:

1. Ensure the deployed system can handle varying loads efficiently.
2. Use load balancers, distributed systems, or containerization tools like Docker and Kubernetes.