Since 2011

RIG
Realistic Infotech Group

Our outcomes are over 5000 trainees.

Get IT Right from RIG

Artificial Intelligence Engineering (Level-1)

Module 1

Learn, Create, and Shine

Digital Space

Content

- Module 1: Introduction to AI and Machine Learning
- Module 2: Linear Algebra, Statistics and Probability for AI
- Module 3: Neural Network Architecture
- Module 4: Building Machine Learning Models
- Module 5: Deep Learning Concepts
- Module 6: Python Data Structure
- Module 7: Data Handling with Pandas and NumPy
- Module 8: Python for AI
- Module 9: Classification AI Project
- Module 10: Prediction AI Project

Artificial Intelligence Engineering
(Level-1)

Module 4: Building Machine Learning Models

Content

- What is Machine Learning
- Types of Machine Learning
- Supervised Learning
- How Supervised Learning Works
- Unsupervised Learning
- How Unsupervised Learning Works
- Reinforcement Learning
- How Reinforcement Learning Works
- How to build Machine Learning Model

Learning Outcomes

- What is Machine Learning: Understand the core concept of machine learning and its role in enabling systems to learn from data and make predictions.
- Types of Machine Learning: Differentiate between supervised, unsupervised, and reinforcement learning paradigms.
- Supervised Learning: Learn how supervised learning uses labeled data for training models to make accurate predictions.
- How Supervised Learning Works: Explore the workflow of supervised learning, including data preprocessing, model training, and evaluation.
- Unsupervised Learning: Understand the purpose of unsupervised learning in discovering hidden patterns and structures in unlabeled data.

Learning Outcomes

- How Unsupervised Learning Works: Gain insight into the working mechanism of clustering, dimensionality reduction, and association rule techniques.
- Reinforcement Learning: Learn about reinforcement learning and its applications in decision-making and control problems.
- How Reinforcement Learning Works: Understand the interaction between agents and environments, including rewards, policies, and value functions.
- How to Build a Machine Learning Model: Develop skills to create a machine learning pipeline, including data preparation, feature selection, and model building.
- Practical Applications: Apply machine learning knowledge to real-world problems across diverse domains such as healthcare, finance, and robotics.

What is Machine Learning?

A field of Artificial Intelligence (AI) that focuses on enabling machines to learn from data and improve their performance over time.

In ML, machines are trained to find patterns, make predictions, or solve tasks by analyzing large datasets.

Traditional Programming vs Machine Learning

Definition and Approach

	Traditional Programming	Machine Learning
Definition	Explicitly coding rules and instructions to solve problems	Building systems that learn patterns from data to make predictions or decisions.
Input	Rules (Logic) + Data -> Program Output	Data + Output (Labeled/ Unlabeled) -> Learn rules (Model)
Output	The program generates outputs based on predefined rules.	The system “learns” rules automatically to generate predictions.

Process

	Traditional Programming	Machine Learning
Rules Creation	Programmers write specific instructions.	Algorithms learn rules from data.
Dependency	Relies on human expertise to write logic	Relies on data and learning algorithms.
Adaptability	Hard to adapt to new situations (requires reprogramming)	Can adapt automatically as new data is provided.

Problem Solving

	Traditional Programming	Machine Learning
Complex Problems	Difficult to handle unstructured data like images or text.	Excels in solving complex problems with data patterns
Rule-based Systems	Suitable for well-defined problems.	Suitable for problems where rules are unknown.

Data Usage

	Traditional Programming	Machine Learning
Data Requirements	Uses predefined rules; does not reply on data.	Requires large amounts of data to learn patterns
Example	Input/ output is static and fixed.	Input/output is dynamic and based on training data.

Errors and Debugging

	Traditional Programming	Machine Learning
Errors	Debugging involves identifying logical errors in code.	Errors depend on data quality and model design.
Performance	If logic is correct, results are deterministic.	Performance depends on training data and model accuracy.

Learning and Improvement

	Traditional Programming	Machine Learning
Learning Process	No learning; fixed instructions.	The model improves its accuracy by learning from data.
Adaption	Requires manual modification of the code.	Learns and adapts automatically to new data.

Importance of Machine Learning

Automation: Machines can perform tasks without constant human intervention.

Data-Driven Decisions: ML models analyze large amounts of data and extract actionable insights.

Improved Accuracy: With sufficient data and good models, machines can outperform humans on specific tasks.

Scalability: ML solutions can process vast amounts of data quickly.

Use Cases of Machine Learning

Healthcare: Diagnosing diseases based on medical images.

E-commerce: Personalized product recommendations.

Finance: Fraud detection in credit card transactions.

Self-Driving Cars: Learning to navigate by analyzing sensor data.

Voice Assistants: Recognizing speech and answering questions (Alexa, Siri).

Types of Machine Learning

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Training a model using labeled data.	Finding patterns in data without predefined labels.	Learning through rewards and penalties.

Popular Use Cases:

Examples include image recognition, spam filtering, recommendation systems, and language translation.

Supervised Learning

Definition	Examples	Popular Algorithms
• Learning from labeled data to make predictions.	• Email Spam Detection (Classification) • House Price Prediction (Regression)	• Linear Regression, Decision Trees, Support Vector Machines (SVM)

Benefits of Supervised Learning

- **Clear Objective**

  - Since data is labeled, the learning process has a clear target.

- **Accurate Predictions**

  - Supervised learning can deliver high accuracy if the data quality is good.

- **Handles Real-World Problems**

  - Many real-world problems like fraud detection, recommendation systems, and medical diagnostics fit well into the supervised learning framework.

- **Easily Interpretable**

  - Results from supervised models are often explainable (e.g., decision trees, linear regression).

- **Improves with Data**

  - More labeled data allows the model to improve and generalize better.

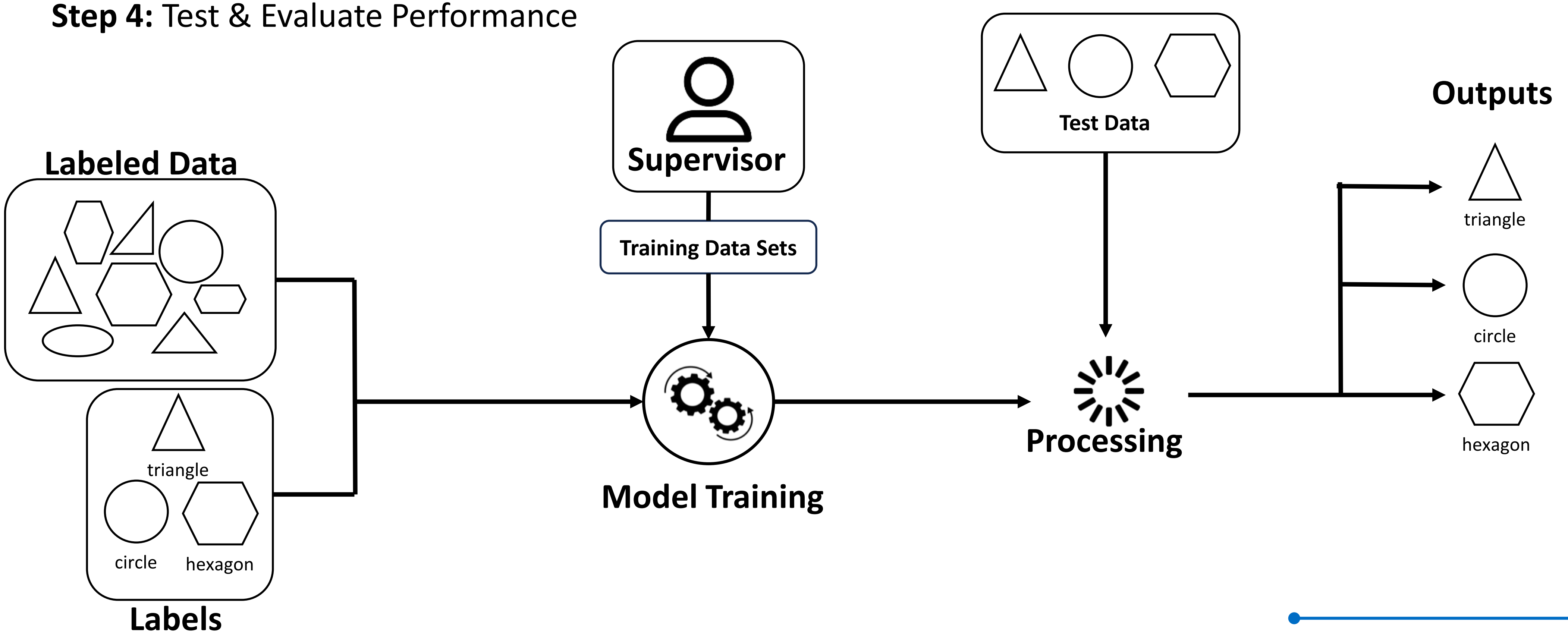
How Supervised Learning Works?

Step 1: Collect Labeled Data

Step 2: Split into Training and Testing Sets

Step 3: Train the Model

Step 4: Test & Evaluate Performance



Steps to Train Supervised Learning Model

Step 1 Define the Problem

Decide if the problem is a regression or classification task.

Step 2 Collect and Prepare Data

Gather labeled data (inputs and outputs)

Size (sq ft)	Location	Price (label)
1200	A	$200,000
1500	B	$250,000

Step 3 Data Preprocessing

- ✓ Clean the Data: Remove missing values and outliers.
- ✓ Normalize or Scale: Scale input features to ensure they are in the same range.
- ✓ Train-Test Split: Split data
- ✓ Training Set: 80% of the data used for training.
- ✓ Testing Set: 20% used for evaluation.

Step 4 Choose the Right Algorithm

- ✓ For Regression: Linear Regression, Polynomial Regression.
- ✓ For Classification: Logistic Regression, Support Vector Machines, Decision Trees.

Step 5 Train the Model

- ✓ Fit the model to the training data.

$$\text{Model} \leftarrow \text{Algorithm}(X_{\text{train}}, Y_{\text{train}})$$

- ✓ The algorithm optimizes parameters to minimize prediction errors.

Step 6 Evaluate the Model

- ✓ Use metrics to measure performance
- ✓ Regression: Mean Squared Error (MSE), R^2.
- ✓ Classification: Accuracy, Precision, Recall, F1-score.

Step 7 Optimize and Tune Hyperparameters

- ✓ Improve the model by tuning parameters like learning rate, model depth, etc.

Step 8 Deploy and Predict

- ✓ Use the trained model to make predictions on new data.

Challenges of Supervised Learning

- **Requires Labeled Data**
  - Obtaining high-quality labeled data can be expensive and time-consuming.
- **Overfitting**
  - The model may perform well on training data but poorly on new data.
  - Solution: Use techniques like cross-validation and regularization.
- **Data Bias**
  - If the training data is biased, the model may produce biased predictions.
- **Scalability**
  - Training on large datasets may require significant computational resources.

Unsupervised Learning

Definition	Examples	Popular Algorithms
• Finding patterns in data without labels.	• Customer Segmentation (Clustering) • Anomaly Detection (Outlier Detection)	• k-Means Clustering, Principal Component Analysis (PCA), Hierarchical Clustering

Benefits of Unsupervised Learning

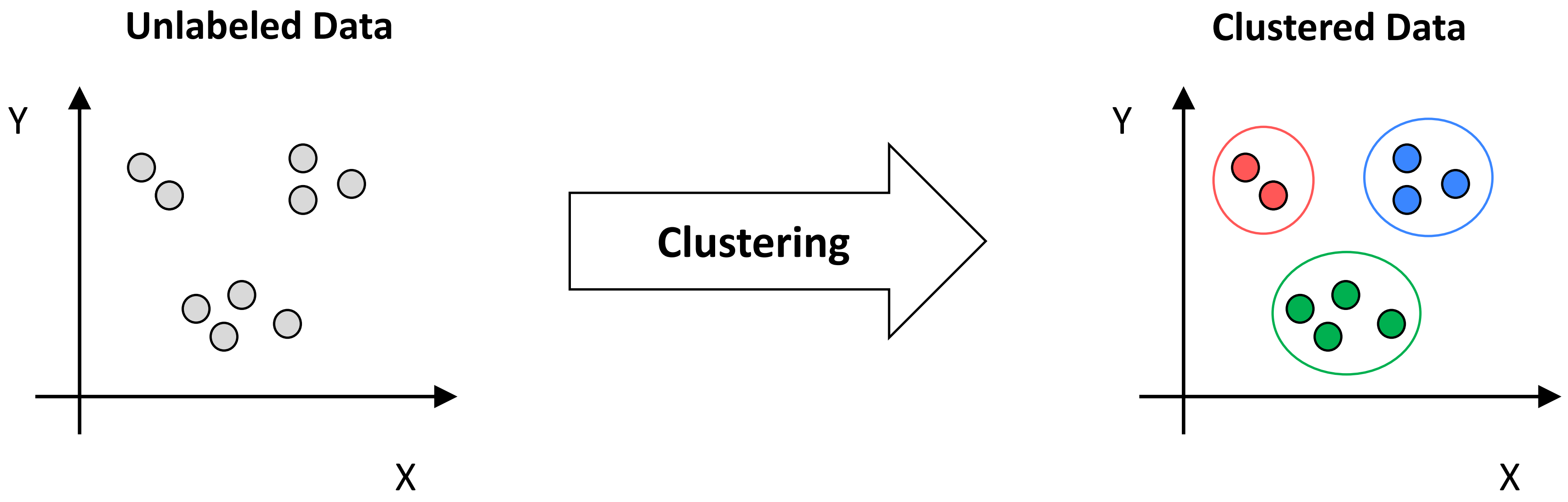
- **No Need for Labeled Data**
  - Unsupervised learning works with unlabeled data, which is cheaper and easier to collect.
- **Identifies Hidden Patterns**
  - It helps discover structures or patterns in the data that may not be obvious.
- **Useful for Large Datasets**
  - Unsupervised learning scales well and is ideal for analyzing large, complex datasets.
- **Feature Reduction**
  - Reduces data complexity while retaining key features, which improves processing efficiency.
- **Insights and Anomaly Detection**
  - Can detect unusual data points (anomalies) in a dataset, like fraud detection or fault detection.

How Unsupervised Learning Works?

Step 1: Input Unlabeled Data

Step 2: Apply Clustering/Dimensionality Reduction

Step 3: Discover Patterns or Groups



Steps to Train Unsupervised Learning Model

Clustering with K-Means

Step 1 Define the Problem

- ✓ Understand the purpose of clustering
- ✓ Group customers into clusters based on their purchasing behavior, income, or spending score

Step 2 Collect and Prepare Data

- ✓ Gather the dataset without labels (no target variable)
- ✓ Customer data, web activity logs, or sensor readings

Step 3 Data Preprocessing

- ✓ Handle Missing Values
- ✓ Remove Outliers
- ✓ Scale/Normalize Features
- ✓ Scale features to ensure they contribute equally to the model
- ✓ Min-Max Scaling or Standardization

Step 4 Choose the Number of Clusters (k)

- ✓ K-Means requires specifying the number of clusters k in advance.
- ✓ **Elbow Method**: Plot the sum of squared distances (inertia) for different values of k and look for the "elbow point" where adding more clusters doesn't significantly reduce the inertia.
- ✓ **Silhouette Score**: Measures how well data points fit within their clusters. A higher score indicates better clustering.

Step 5 Train the Model

- ✓ Initialize k cluster centroids (randomly).
- ✓ Assign each data point to the nearest cluster centroid (using Euclidean distance)
- ✓ Update the centroids: Calculate the mean position of all points in each cluster.
- ✓ Repeat until centroids stop changing significantly or a predefined number of iterations is reached.

Assign points to the closest centroid.

Update centroids:

$$C_i = \frac{1}{n}\sum_{j=1}^{n} X_j$$

- C_i: New centroid for cluster i.
- X_j: Points in cluster i.
- Repeat until convergence.

Step 6 Evaluate the Clusters

- ✓ Measure the quality of clusters using metrics
- ✓ **Inertia (Within-Cluster Sum of Squares)**: Lower values indicate tighter clusters.
- ✓ **Silhouette Score**: Measures how similar points are to their own cluster compared to other clusters.
- ✓ Visualize the clusters using scatter plots (2D or 3D).

Step 7 Interpret and Analyze Results

- ✓ Examine the clusters to understand their meaning.
- ✓ Cluster 1 → Low income, low spending.
- ✓ Cluster 2 → High income, high spending.
- ✓ Use insights to make decisions, such as targeted marketing or anomaly detection.

Step 8 Optimize the Model

- ✓ Test different values of k to identify the optimal number of clusters.
- ✓ Use dimensionality reduction (PCA) if the dataset has high dimensionality to improve efficiency.

Step 9 Deployment

- ✓ Deploy the model to group incoming data (new customers).
- ✓ Integrate into applications for continuous clustering and decision-making.

Reinforcement Learning

Definition

- Learning through rewards and penalties in a trial-and-error fashion.

Examples

- Game AI (AlphaGo)
- Robotics (Path Optimization)

Key Concepts

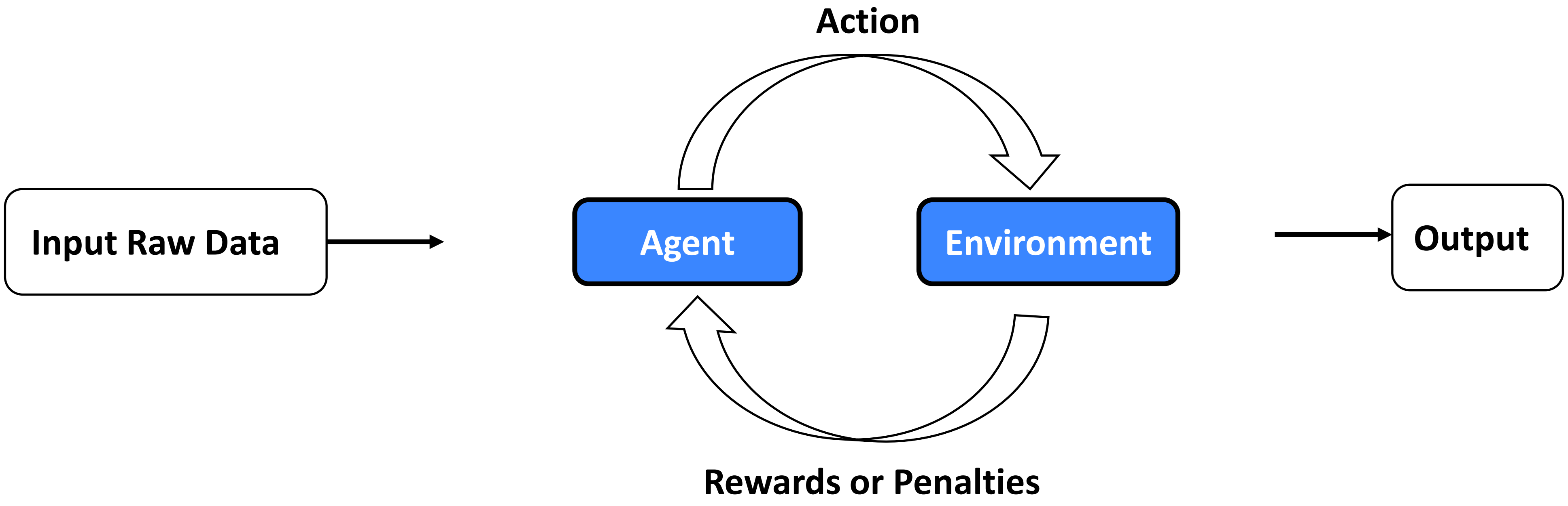
- Agent, Environment, Rewards, Actions

How Reinforcement Learning Works?

Step 1: Agent interacts with the environment.

Step 2: Receives rewards or penalties.

Step 3: Learns to maximize rewards over time.

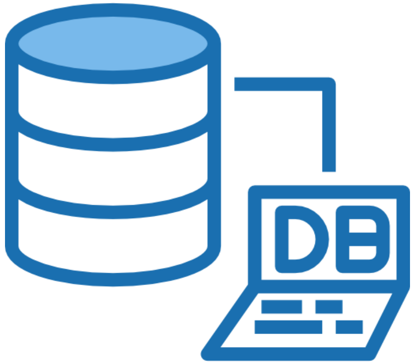


How to build Machine Learning Model?

Understanding the Machine Learning Workflow

Step 1: Data Collection

Gathering data from sources like **CSV files**, **databases**, or **web scraping**.



Step 2: Data Preprocessing

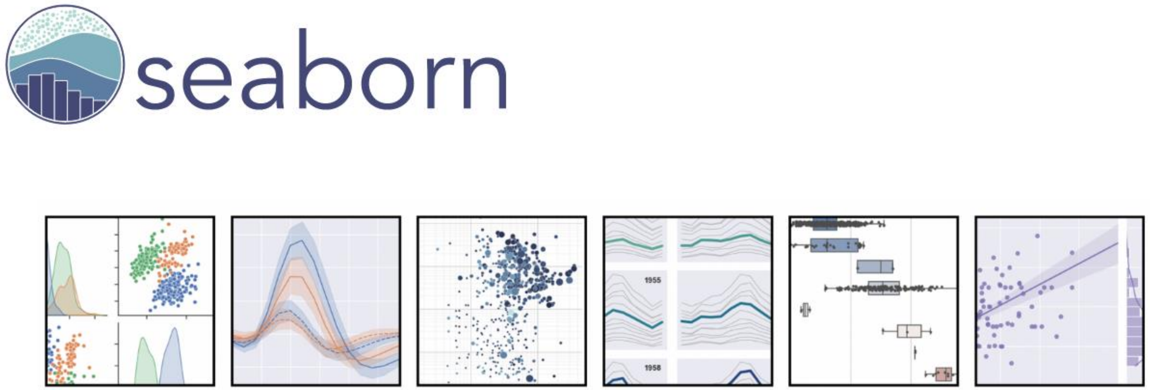
- Handling missing values, normalizing data, and converting categorical variables.
- Tools: pandas for data manipulation.

- Handling missing values
- Normalizing Data
- Converting categorical variables

```
import pandas as pd
```

Step 3: Exploratory Data Analysis (EDA)

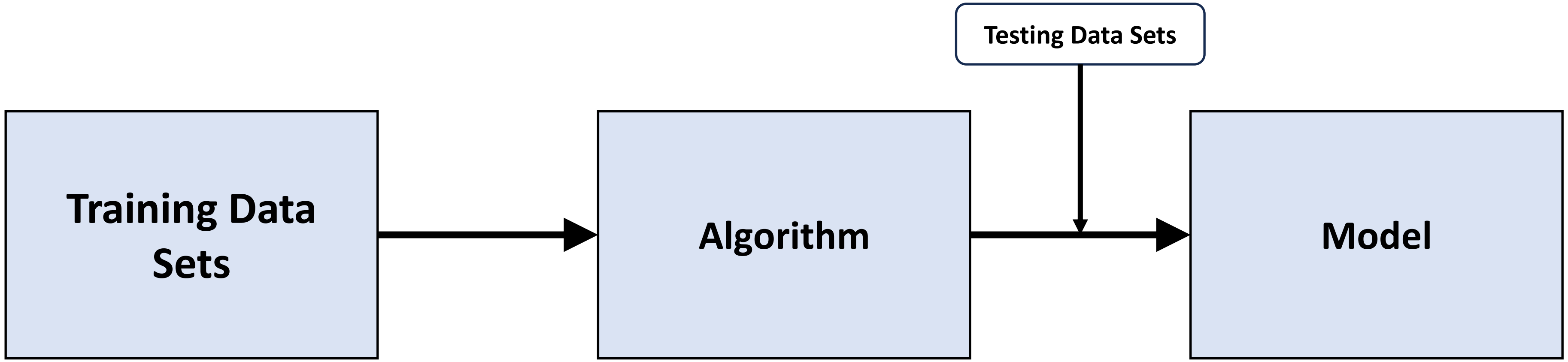
- Using **matplotlib** and **seaborn** for visualizing data patterns.




```python
import matplotlib.pyplot as pd
import seaborn as sns
```

Step 4: Model Building

- Splitting data into training and testing sets.
- Choosing the right algorithm for your problem.



Step 5: Model Evaluation

- Metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

True Positive (TP): Correctly predicted positive cases
True Negative (TN): Correctly predicted negative cases
False Positive (FP): Incorrectly predicted positive cases
False Negative (FN): Incorrectly predicted negative cases

Equations:

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

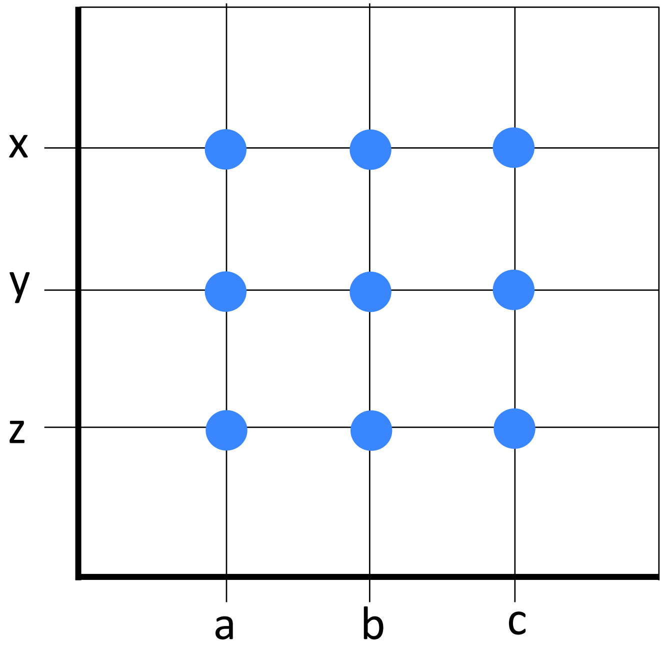
$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

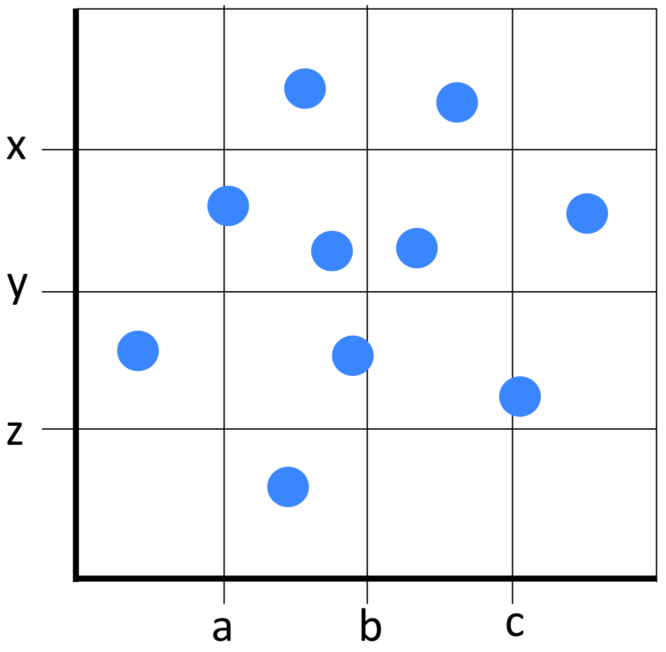
$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Step 6: Hyperparameter Tuning

➢ Techniques like **Grid Search** and **Random Search**.



Grid Search



Random Search

Step 7: Model Deployment

- Exporting models using joblib or pickle.
- Deploying models using Flask or FastAPI.

```
from joblib import dump, load

import pickle
```

```
from flask import Flask, request,
josonify

from fastapi import FastAPI
```

RIG

Realistic Infotech Group

Realistic Infotech Group
IT Training & Services
No.79/A, First Floor
Corner of Insein Road and
Damaryon Street
Quarter (9), Hlaing Township
Near Thukha Bus Station
09256675642, 09953933826
http://www.rig-info.com