Lesson Plan — Supervised Learning Algorithms

Duration: 60 minutes
Audience: Beginner to intermediate students in Machine Learning / Data Science
Learning Objectives:
By the end of this lesson, students will be able to:

1. Explain the concept of supervised learning.
2. Differentiate between classification and regression tasks.
3. Describe common supervised learning algorithms and their use cases.
4. Train and evaluate a basic supervised learning model.

---

0 – 5 min | Introduction & Context

- **Hook:** Start with a relatable example:
  "Imagine you're building an app to predict house prices based on location, size, and features — how could we do that using past data?"
- **Definition:** Introduce *Supervised Learning* as training a model with labeled data (inputs + known outputs).
- **Why it matters:** Highlight use cases — spam detection, medical diagnosis, credit scoring.

---

5 – 15 min | Core Concepts

- **Input vs Output:** Explain *features* and *labels*.
- **Two main types:**
  1. **Classification** — Predict discrete categories (e.g., spam / not spam).
  2. **Regression** — Predict continuous values (e.g., house price).
- **Examples Table:**

Problem	**Type**	**Example Output**
Email spam detection	Classification	Yes / No
Loan default risk	Classification	Low / Medium / High
House price prediction	Regression	$250,000
Predict temperature	Regression	28.5°C

---

15 – 25 min | Common Algorithms Overview

- **k-Nearest Neighbors (k-NN):** Simple, distance-based predictions.

- **Decision Trees:** Rule-based splitting of data.
- **Linear Regression:** Fits a straight line for numeric prediction.
- **Logistic Regression:** Classification using probability thresholds.
- **Support Vector Machines (SVM):** Finds decision boundaries in high-dimensional space.

💡 Use **visual diagrams** to show how each algorithm works.

---

25 – 40 min | Hands-On Demo

- **Tool:** Google Colab or Jupyter Notebook.
- **Dataset:** Iris dataset (classification) or Boston Housing dataset (regression).
- **Steps:**
    1. Load dataset (sklearn.datasets.load_iris()).
    2. Split into train/test sets.
    3. Train a simple Decision Tree.
    4. Evaluate accuracy with accuracy_score.
- Encourage students to change hyperparameters and observe the effect.

---

40 – 50 min | Evaluation & Challenges

- Explain **train/test split** and **cross-validation**.
- Introduce metrics: accuracy, precision, recall, RMSE.
- Discuss **overfitting** and **underfitting** with simple graphs.

---

50 – 57 min | Mini-Activity

- Give students **3 short scenarios** and ask them to:
    1. Identify if the task is classification or regression.
    2. Suggest a suitable supervised learning algorithm.
- Example:
    - Predict customer churn → Classification → Logistic Regression / Decision Tree.
    - Predict exam score → Regression → Linear Regression.

---

57 – 60 min | Wrap-Up

- Summarize key takeaways: definition, types, algorithms, evaluation.

- Q&A session.
- Optional homework: Try training a different supervised algorithm on the same dataset.

📌 Materials Needed

- Slides with definitions, examples, and diagrams.
- A Jupyter Notebook/Colab demo script.
- Printed mini-activity sheet.