# UNIT 5 → Scalable Machine Learning

### **Semi - Supervised Learning :**

**SKLearn has two methods to build a Semi-Supervised Model.**

1. Label Propagation
2. Label Spreading

**Steps for Building a Semi-Supervised Model :**

1. Build a Supervised Model as all the examples in the dataset are Labeled.
2. Label means y-value/target value.(-1).
3. SKLearn requires (-1) value as target for those examples/rows/instances that have no Labels.
4. Now the dataset contains 250 Labeled data points & 250 unlabeled data points.
5. **Label Propagation** :
   * 1. Supervised learning Accuracy = 84%
     2. Semi-Supervised learning Accuracy = 85.6%
6. **Label Spreading** :
   * 1. Supervised learning Accuracy = 84%
     2. Semi-Supervised learning Accuracy = 85.39%
7. Note : Semi Supervised is close to Supervised Model.

**Implementation of Semi-supervised Model using SKLearn :**

[Building Semi-supervised model using Various Methods](https://colab.research.google.com/drive/12Wiu2DOj3oMgWOZi-q5wBtL52r018xC4?usp=sharing)

### **Scalable Machine Learning :**

1. Apply Machine learning to very large datasets.
2. So far, we have used datasets of small to medium sizes where the dataset can fit into your main memory space that can be done on your pc or single systems where you use CPU/GPUs.
3. For Scalable machine learning algorithms, the available single systems may not support.
4. **Applications of Scalable Machine Learning** :
   1. Autonomous Driving
   2. Gaming
   3. Machine Translation
5. **Challenges Are :**
   1. **Data Handling** :Entire dataset may not fit into main memory for training.
   2. Training on large datasets requires a lot of training time & more computing power where single processors may not handle the processing.
6. **Approaches :** 
   1. **Vertical Scaling** : Enhancing computing power & main memory storage like super computers. But such devices are expensive.
   2. **Horizontal Scaling :**
      1. You have a number of systems connected through a network called Distributed Systems.
      2. Each system is called a node of the network in a distributed environment.
      3. Each node stores some part of the dataset and also participates in the computation of the training process.
      4. All nodes together participate in completing the training process.
      5. Distributed systems perform distributed storage & distributed computing. Such systems are less expensive and still able to do the job.
      6. A popular distributed framework is **Hadoop**. It uses two methods ;
         1. **Data Storage - HDFS** : Data is divided into smaller chunks
         2. **Computing Environment - Map Reduce** :
            1. Uses two functions called map & reduce.
            2. Same map is applied on each of the chunks in different nodes.
            3. Then the results of this map function are combined using reduce function to get the final output.
   3. For standard machine learning algorithms are applied on small/medium datasets on a single system & maybe multiple cores.
   4. Eg: SKLearn is a framework for standard Machine Learning, TensorFlow, Kera, PyTorch.
   5. Scalable Machine Learning Frameworks : Apache Spark.

### **Online Learning :**

* Entire dataset is not available for learning.
* Dataset with few data points are initially available and more data points keep coming. So the model has to evolve over time.
* You have to keep updating the model as the new data points keep arriving.
* Uses Stochastic gradient descent Algorithm.
* Eg: E-Commerce Websites use online learning for recommendations, Social media also use online learning for predicting the next probable post.
* Gradient Descent Algorithm :

1. **Gradient Descent Algorithm** : Use the entire dataset to update the parameters.
2. **Batch Gradient Descent Algorithm** : Use a subset of the dataset to update the parameters & repeats till the whole dataset is processed.
3. **Stochastic Gradient Descent Algorithm** : Parameters are updated after each data point is processed.

| **Offline Learning** | **Online Learning** |
| --- | --- |
| 1. Entire dataset is available for training. 2. Model is less complex. 3. Patterns/ Structure in data are constant. 4. Easy to implement. 5. Eg : SKLearn, TensorFlow, Pytorch are frameworks. | 1. Only few data points are initially available & data   keeps arriving overtime.   1. The model keeps evolving as data points arrive. 2. Patterns in data keep changing/emerging. 3. Difficult to implement and manage. 4. Eg : StreamDM, ScikitMultiflow |

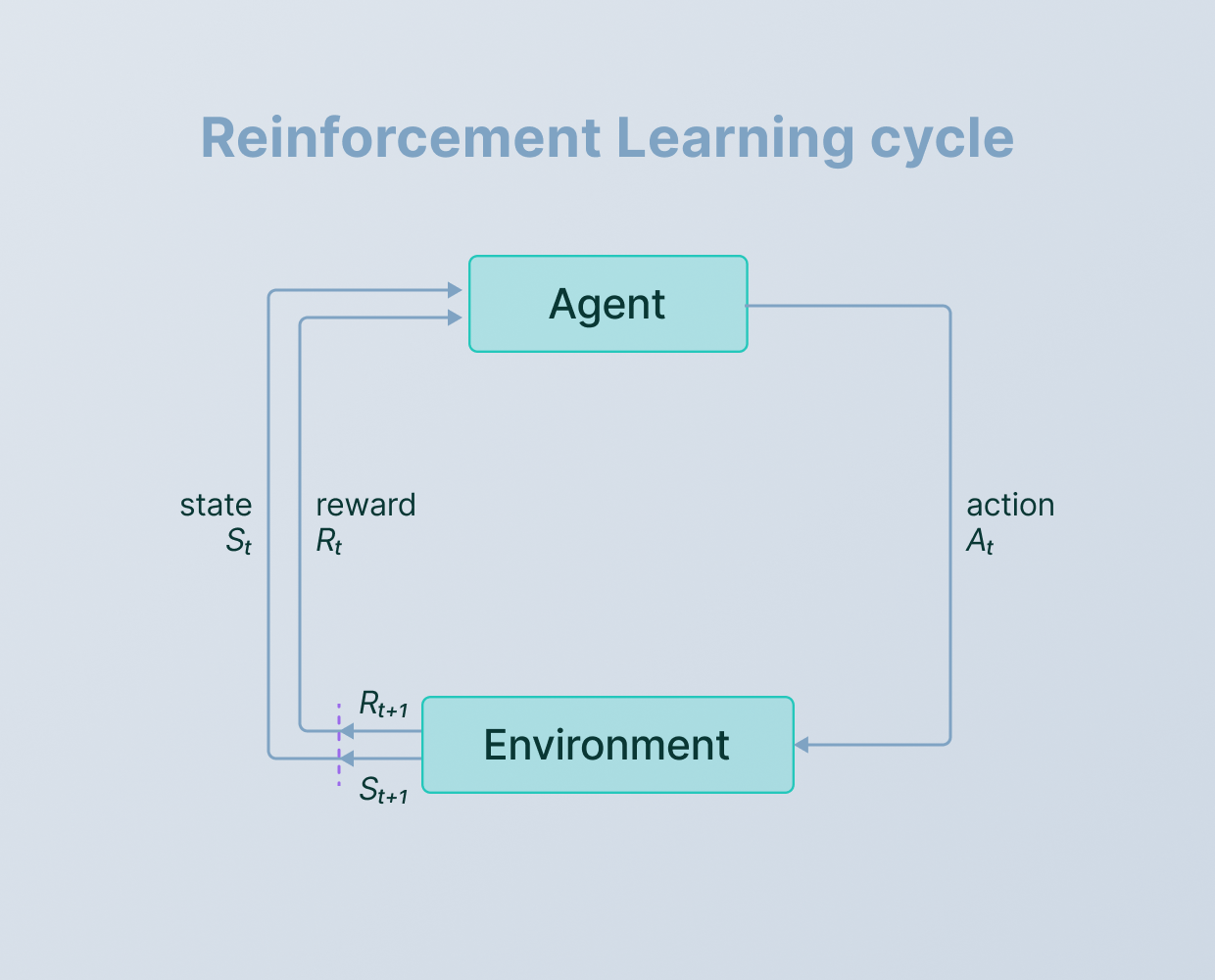
All the Machine Learning Algorithms can be broadly classified into;

1. **Supervised Learning Algorithms** : Also called guided learning. Dataset has features as well as targets. Mainly used for predictive models. Given a new datapoint, you can predict the target value.
   1. Regression
   2. Classification
2. **Unsupervised Learning Algorithms** : Also called as unguided learning. Dataset has only features but no target. This is used for finding patterns/groups/structures in the data.
   1. Clustering
   2. Dimensionality Reduction
3. **Semi-Supervised Learning Algorithms** : Only some of the data points have target values but majority of the data points are unlabelled i.e., no target values. You have to predict the labels for the unlabelled data points.
4. **Reinforcement Learning** :

You learn by experience.

### **Reinforcement Learning :**

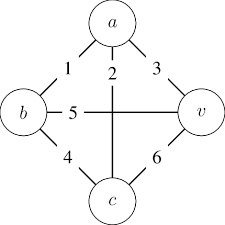
1. You learn by experience.
2. Eg. A child learning to walk. Child tries to learn to walk by trial and error.
   1. Here, the child is an **Agent**.
   2. **Environment** where the child is making an attempt to learn.
   3. **Action** is trying to walk.
   4. **Reward** is encouragement the agent gets in the learning process.
3. **Steps** :
   1. Reinforcement learning starts from the initial state () of the environment.
   2. Based on , the agent takes the action .
   3. Environment gets updated i.e., going to the next stage .
   4. Based on action, the agent gets the reward .
   5. The above steps are repeated until the task completes.



The agent tries to maximize his rewards in this entire process and complete the task.

1. **Elements of Reinforcement Learning** :
2. **Agent** : Who is learning by trial and error. Takes action based on the state of the environment.
3. **Environment** : Where the agent learns and takes actions.
4. **Reward** : Based on action taken to decide on the next step.
5. **Action** : The steps the agent takes to complete the process.
6. **Policy** (**ℼ**) : The approach agent uses to take the next action.
7. **Value** (V) : The consequence the agent gets.

5. Agent uses **Markov decision process** in



Task is to move from A→V.

Possible paths are :

A→V = 3

A→C→V = 8

A →B→C→ V = 11.

Agent chooses the path that gains maximum points i.e, **A→V is 11**.

6. Agent uses the following policies to take the action :

* **Exploration** : Agent explores all the possible actions to maximize the rewards.
* **Exploitation** : Agent uses the knowledge he has already gained to perform an action.