**Unsupervised Learning**

**What is Unsupervised Learning?**

Unsupervised learning is a type of machine learning where the algorithm analyzes data without needing labeled outputs. Unlike supervised learning, where you have a target variable to guide the model, unsupervised learning focuses on finding hidden patterns or structures within the data.

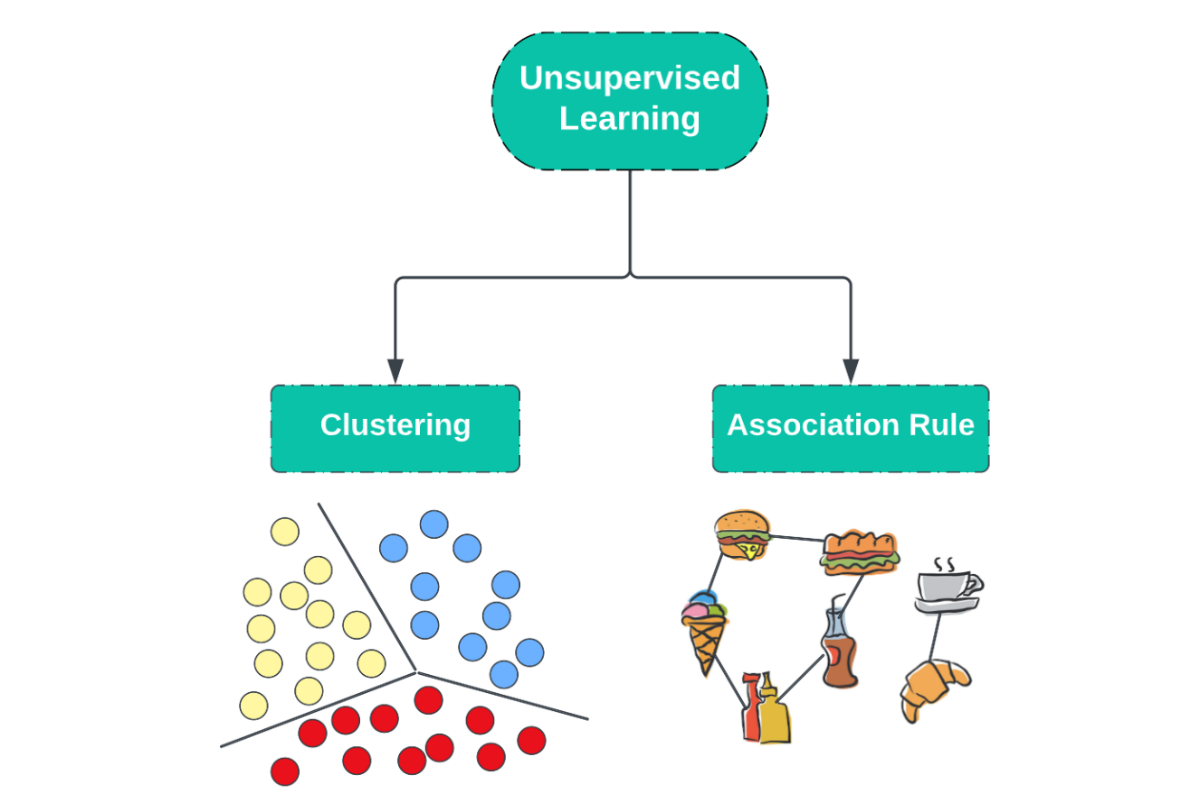
For example, unsupervised learning can be used to:

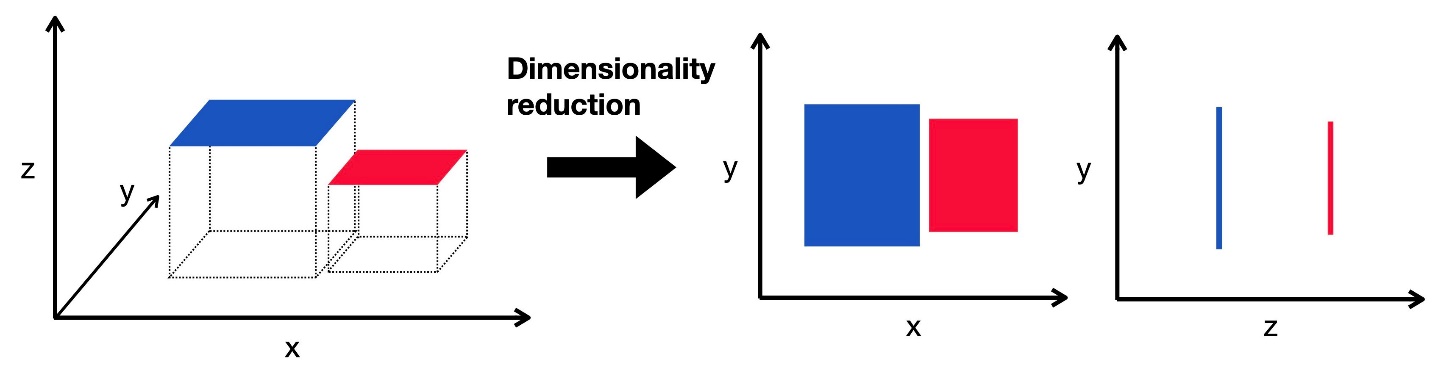
* Group similar images together.
* Identify patterns in customer behaviors.
* Segment geographical areas on a map based on specific features.

**Types of Unsupervised Learning**

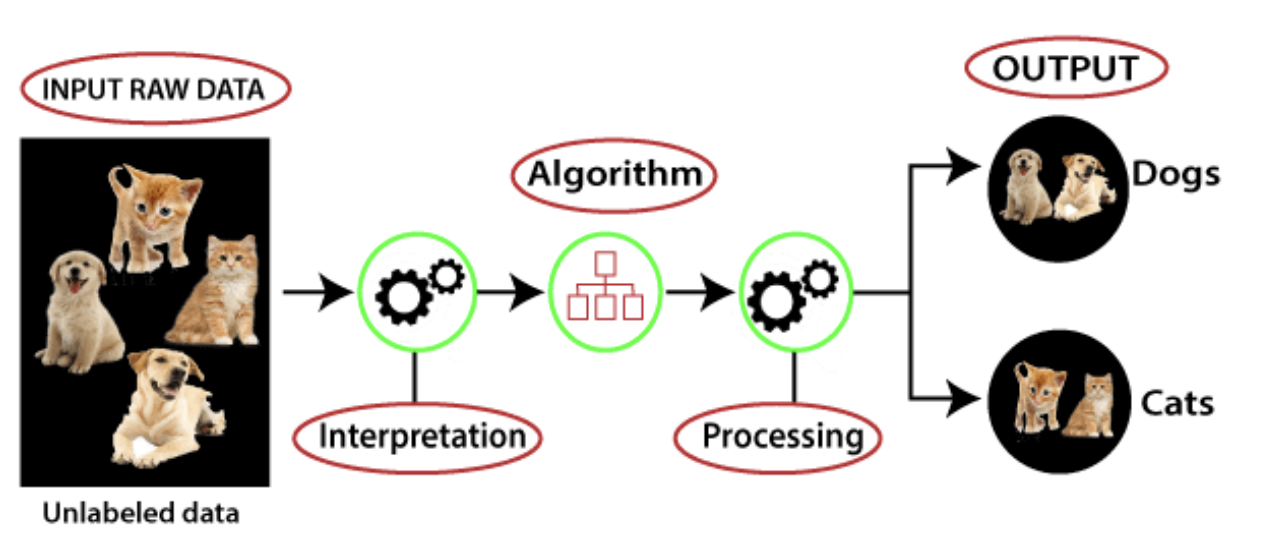
Unsupervised learning can be broadly categorized into three types:

1. **Clustering:** Grouping similar data points together based on their features.
2. **Association:** Discovering relationships between items in a dataset, often used in market basket analysis.
3. **Dimensionality Reduction:** Reducing the number of variables in a dataset while keeping its essential information, like using Principal Component Analysis (PCA).





**How Unsupervised Learning works?**

****

The diagram illustrates the process of unsupervised learning, where an algorithm identifies patterns and structures in unlabeled data. Here's the step-by-step explanation:

1. **Input Raw Data**: The process begins with raw, unlabeled data (e.g., images of cats and dogs in this case). No prior labels or classifications are provided for the data.
2. **Algorithm**:
   * **Interpretation**: The algorithm analyzes the input data to find inherent patterns or similarities within the dataset. For example, it might group images based on shared features like shape, color, or texture.
   * **Processing**: Using clustering or dimensionality reduction techniques, the algorithm organizes the data into meaningful categories or groups.
3. **Output**: The algorithm outputs clusters or categories of data. In this example, it separates the input data into two groups: "Dogs" and "Cats," based on the patterns it identified in the unlabeled images.

This method is fundamental in tasks such as customer segmentation, anomaly detection, and image categorization, where labeled data may not be available.

**Advantages:**

* Useful for exploring unknown patterns in data.
* Requires no labeled data.

**Disadvantages:**

* Results may be less accurate since there is no labeled reference.
* Difficult to evaluate the quality of output.

**Clustering in Detail**

**What is Clustering?**

Clustering is the process of grouping similar objects into sets called clusters. The goal is to have objects in the same cluster be more similar to each other than to those in other clusters. For instance, you might group animals based on their physical features, or customers based on their buying patterns.

**Common Clustering Algorithms**

Some of the most widely used clustering algorithms are:

1. **K-Means Clustering:** This algorithm divides data into a set number of clusters. You choose the number of clusters in advance.
2. **Hierarchical Clustering:** It creates a tree-like structure of clusters, useful for understanding relationships between clusters.
3. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** It groups together points that are close to each other and separates outliers or noise.

**Association in Detail**

**What is Association?**

Association learning identifies interesting relationships or patterns in large datasets. It is commonly used in market basket analysis, where it helps discover which products are frequently bought together.

For instance, if customers often buy bread and butter together, the model might find that these items are associated and suggest them as a pair to future shoppers.

**Example:**

* If a customer buys bread, they are likely to buy butter as well. These relationships are discovered using association rules like Apriori or FP-Growth.

**Applications:**

* Recommending products in online stores.
* Understanding purchasing habits of customers.
* Optimizing inventory based on buying patterns.

**Practical Activity 1: Group Similar Images**

**Objective:**

Use clustering to group similar images into categories.

**Steps:**

1. **Requirements:** A computer with Python installed.
2. **Dataset:** Download a folder of 20-30 images (e.g., animals, vehicles, fruits).
3. **Instructions:**
   * Install the necessary libraries: pip install numpy pandas matplotlib scikit-learn.
   * Load the images using Python.
   * Apply the K-Means clustering algorithm to group the images into clusters.

**Code Example:**

from sklearn.cluster import KMeans

from skimage.io import imread

import numpy as np

import os

# Load images

folder = 'images/'

data = []

for file in os.listdir(folder):

img = imread(os.path.join(folder, file)).flatten()

data.append(img)

data = np.array(data)

# Apply K-Means

kmeans = KMeans(n\_clusters=3, random\_state=42).fit(data)

print("Cluster Labels:", kmeans.labels\_)

**Outcome:**

You’ll be able to see which images belong to the same cluster based on their visual features.

**Practical Activity 2: Create a Music Playlist**

**Objective:**

Use clustering to group songs based on characteristics like tempo and rhythm.

**Steps:**

1. **Requirements:** A computer with Python and a dataset of songs (you can either use an online music dataset or create your own CSV file with features like tempo, beats per minute, etc.).
2. **Instructions:**
   * Install the required libraries: pip install pandas scikit-learn matplotlib.
   * Load the dataset.
   * Apply hierarchical clustering to group songs based on their features.

**Code Example:**

import pandas as pd

from sklearn.cluster import AgglomerativeClustering

import matplotlib.pyplot as plt

# Load dataset

data = pd.read\_csv('music\_features.csv')

features = data[['tempo', 'bpm']]

# Apply Hierarchical Clustering

clustering = AgglomerativeClustering(n\_clusters=3).fit(features)

data['Cluster'] = clustering.labels\_

# Visualize clusters

plt.scatter(data['tempo'], data['bpm'], c=data['Cluster'])

plt.xlabel('Tempo')

plt.ylabel('Beats per Minute')

plt.title('Music Clusters')

plt.show()

**Outcome:**

You will see how clustering can help create personalized playlists based on similar music features.

**Practical Activity 3: Segment Customers**

**Objective:**

Use clustering to identify customer segments based on their spending behavior.

**Steps:**

1. **Requirements:** A computer with Python and a customer dataset (e.g., number of purchases, amount spent).
2. **Instructions:**
   * Install the required libraries: pip install pandas matplotlib seaborn scikit-learn.
   * Load the dataset.
   * Apply DBSCAN to identify clusters of customers.

**Code Example:**

from sklearn.cluster import DBSCAN

import pandas as pd

import matplotlib.pyplot as plt

# Load dataset

data = pd.read\_csv('customer\_data.csv')

features = data[['purchases', 'amount\_spent']]

# Apply DBSCAN

clustering = DBSCAN(eps=3, min\_samples=2).fit(features)

data['Cluster'] = clustering.labels\_

# Visualize clusters

plt.scatter(data['purchases'], data['amount\_spent'], c=data['Cluster'])

plt.xlabel('Number of Purchases')

plt.ylabel('Amount Spent')

plt.title('Customer Segments')

plt.show()

**Outcome:**

By clustering customer data, businesses can better understand spending patterns and tailor their services accordingly.

**Conclusion**

* Unsupervised learning helps discover hidden patterns in data without needing labeled outputs.
* Clustering is a valuable technique for grouping similar objects based on their features.
* Association helps identify relationships between items in datasets, such as products often bought together.
* Unsupervised learning is used in practical applications like image categorization, music recommendation, and customer segmentation.