University Of Canberra

WEEK12

Pattern Recognition and Machine Learning

Created by Elane Peng

Objectives

- Understand the core concepts and types of unsupervised learning. Distinguish unsupervised learning from supervised learning, and identify its main categories such as clustering, dimensionality reduction, and anomaly detection.
- Explain and apply foundational algorithms. Describe the working principles of key algorithms like K-Means clustering, PCA, and autoencoders, and discuss their strengths, limitations, and typical use cases.

Unsupervised Learning

1. What is unsupervised learning, and how does it differ from supervised learning?

Unsupervised learning is a type of machine learning where the model is trained on data without labeled outputs. The goal is to find hidden patterns, structures, or relationships in the data.

Unsupervised Learning

1. What is unsupervised learning, and how does it differ from supervised learning?

Key differences from supervised learning:

Aspect	Supervised Learning	Unsupervised Learning
Data	Labeled (input-output pairs)	Unlabeled
Goal	Predict labels or outputs	Find structure or clusters
Common Algorithms	Linear regression, SVM, decision trees	K-Means, Hierarchical clustering, PCA, Autoencoders
Example	Predicting house prices	Segmenting customers based on purchase behavior

Unsupervised Learning

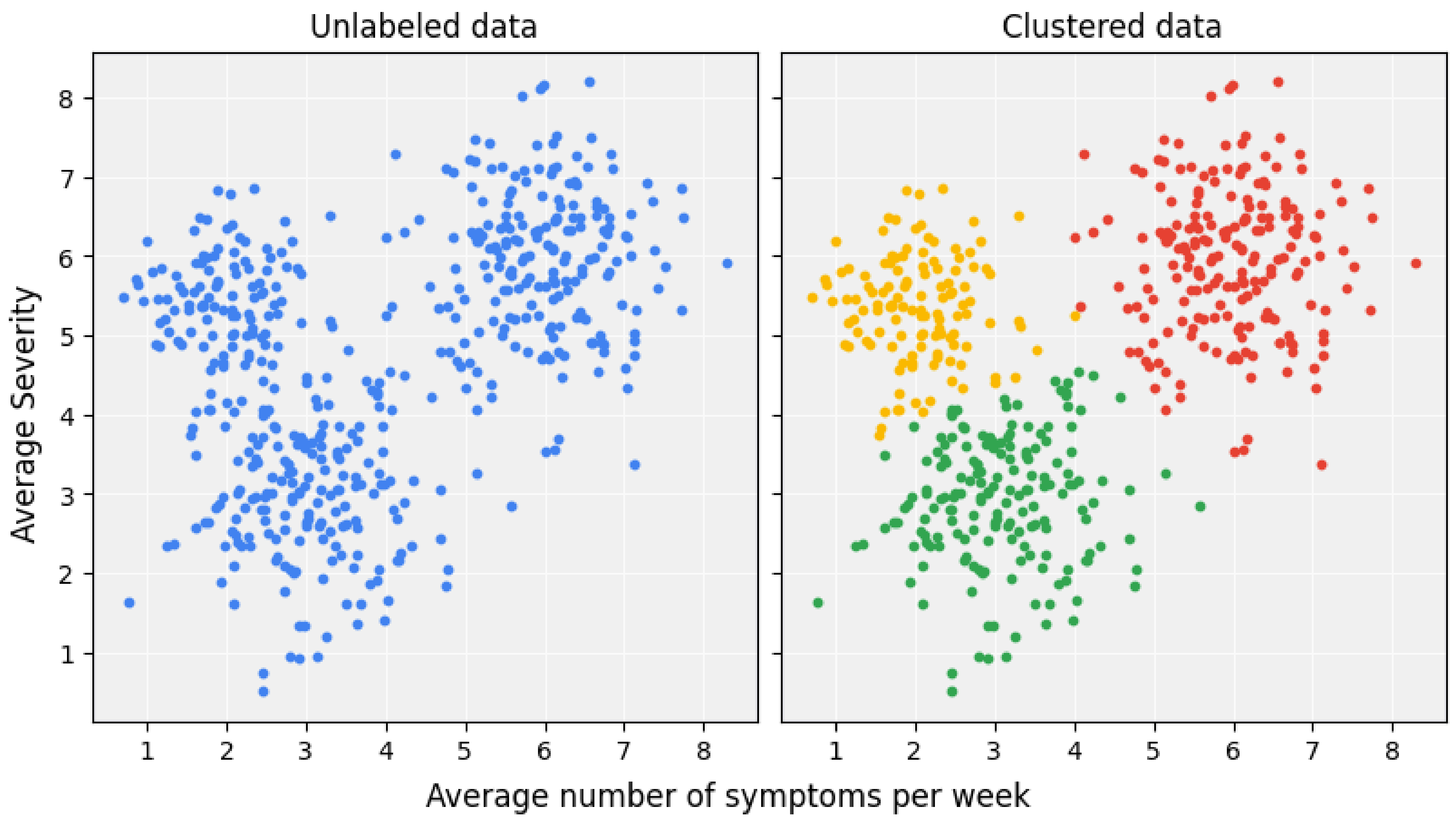
2: What are the main types of unsupervised learning?

1. Clustering:

Groups data points into clusters based on similarity.

Algorithms: K-Means, DBSCAN, Hierarchical clustering

Example: Customer segmentation in marketing

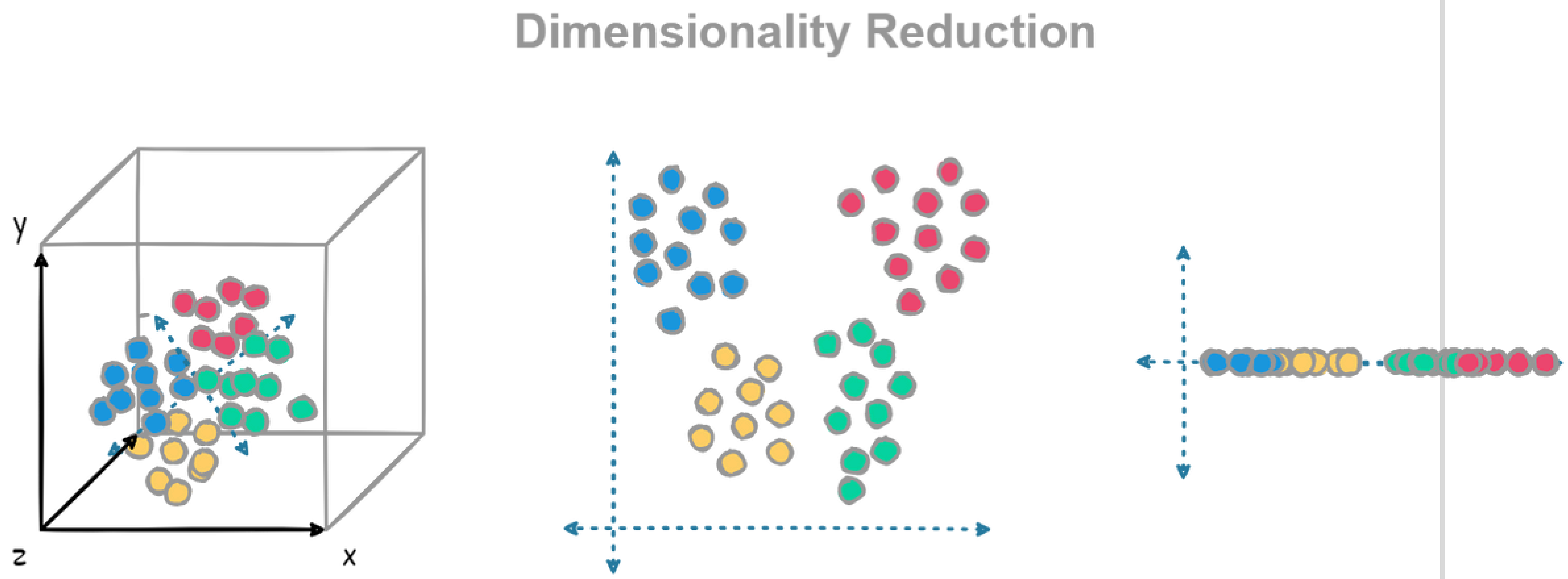


Unsupervised Learning

2: What are the main types of unsupervised learning?

2. Dimensionality Reduction: Reduces data features while preserving structure.

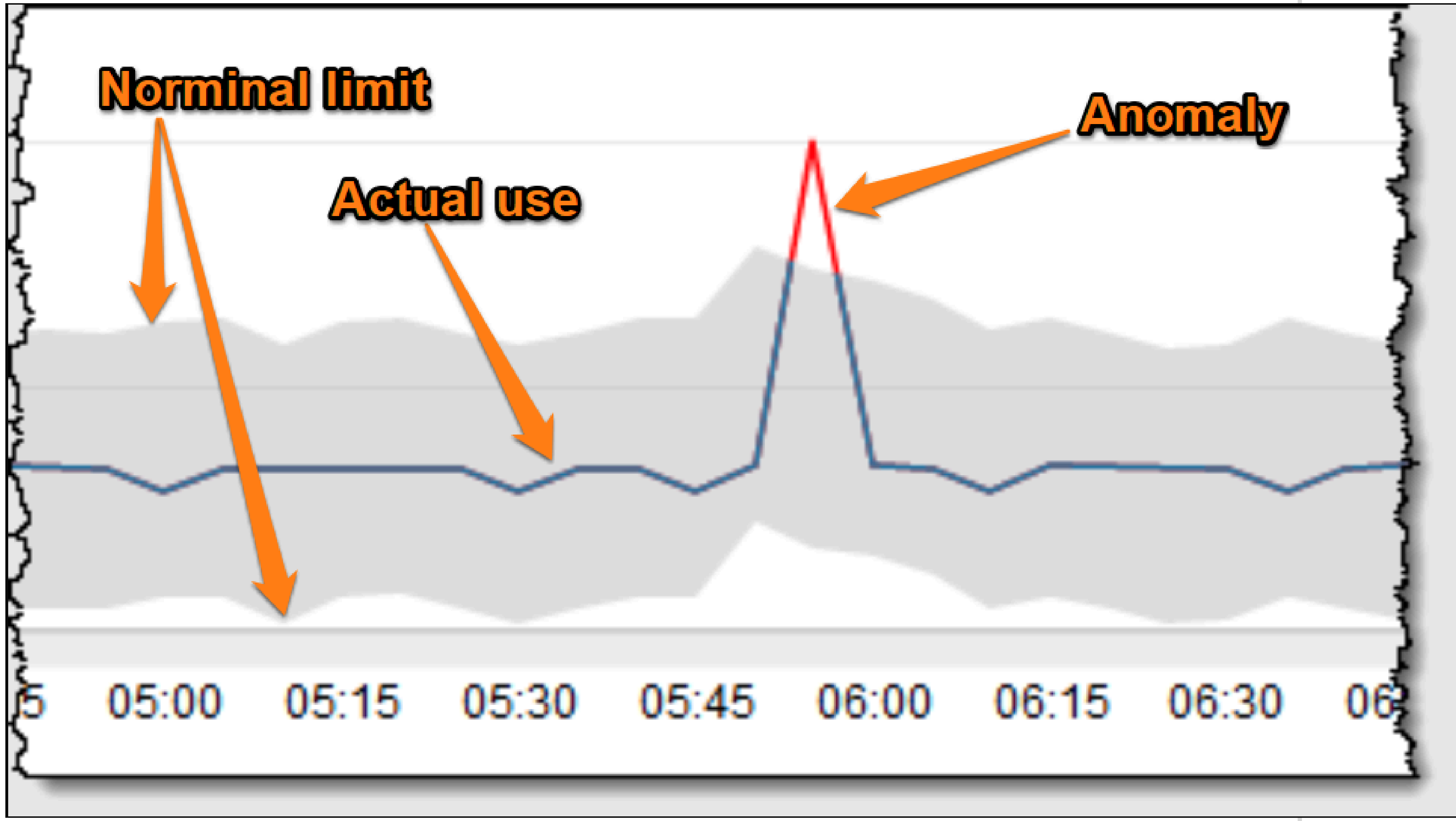
Algorithms: PCA, LDA, t-SNE
Example: Visualizing high-dimensional data, compressing images



Unsupervised Learning

2: What are the main types of unsupervised learning?

3. Anomaly Detection: Identifies unusual or outlier data points.
Algorithms: One-Class SVM, Isolation Forest
Example: Fraud detection, network intrusion detection



Unsupervised Learning

Question 3: Explain K-Means clustering. What are its main limitations?

K-Means is an algorithm that partitions data into K clusters by minimizing the variance within each cluster.

fig 1: before applying k-means clustering

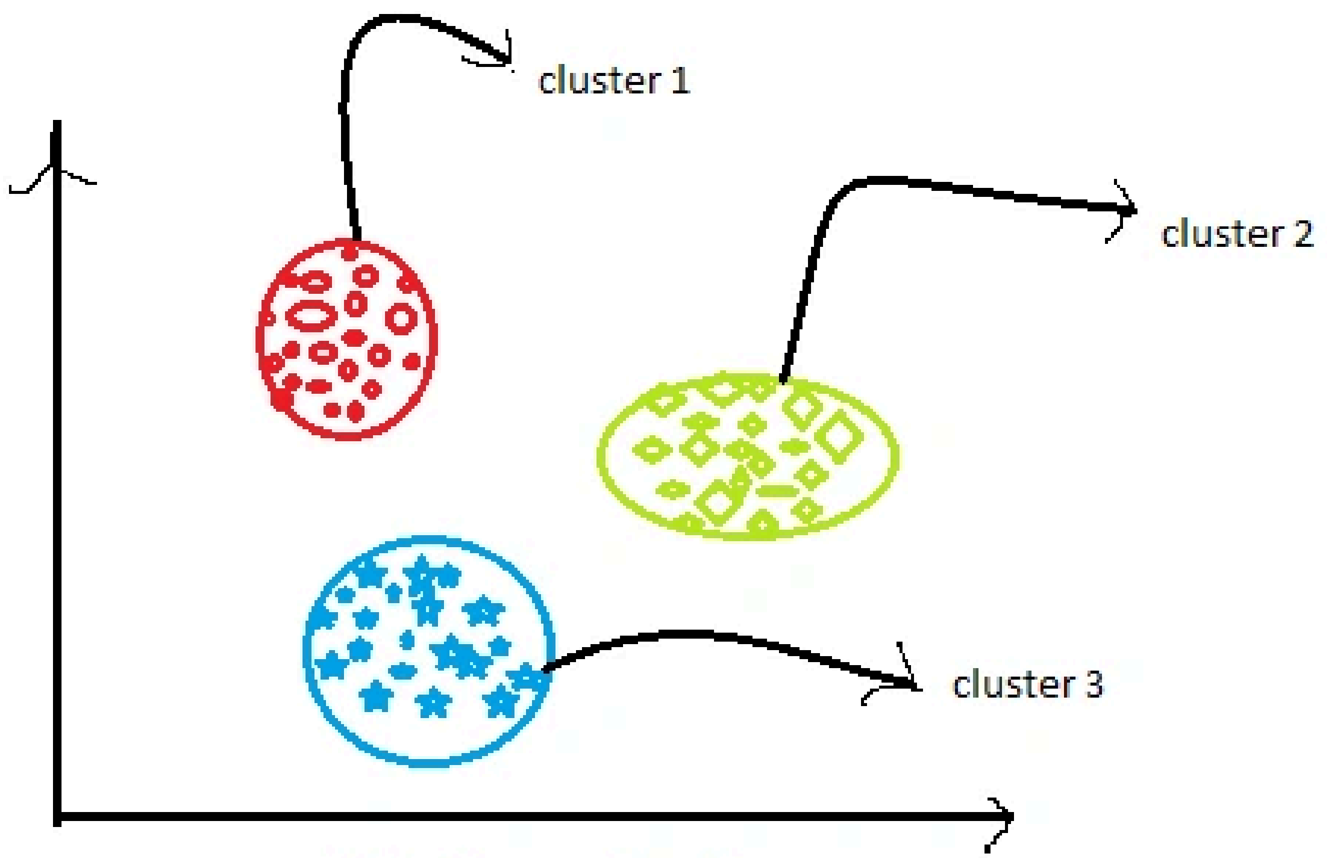


fig 2: After applying K-means clustering

Unsupervised Learning

Question 3: Explain K-Means clustering. What are its main limitations?

Steps:

1. Initialize K centroids randomly.
2. Assign each data point to the nearest centroid.
3. Update centroids as the mean of points in the cluster.
4. Repeat steps 2–3 until convergence.

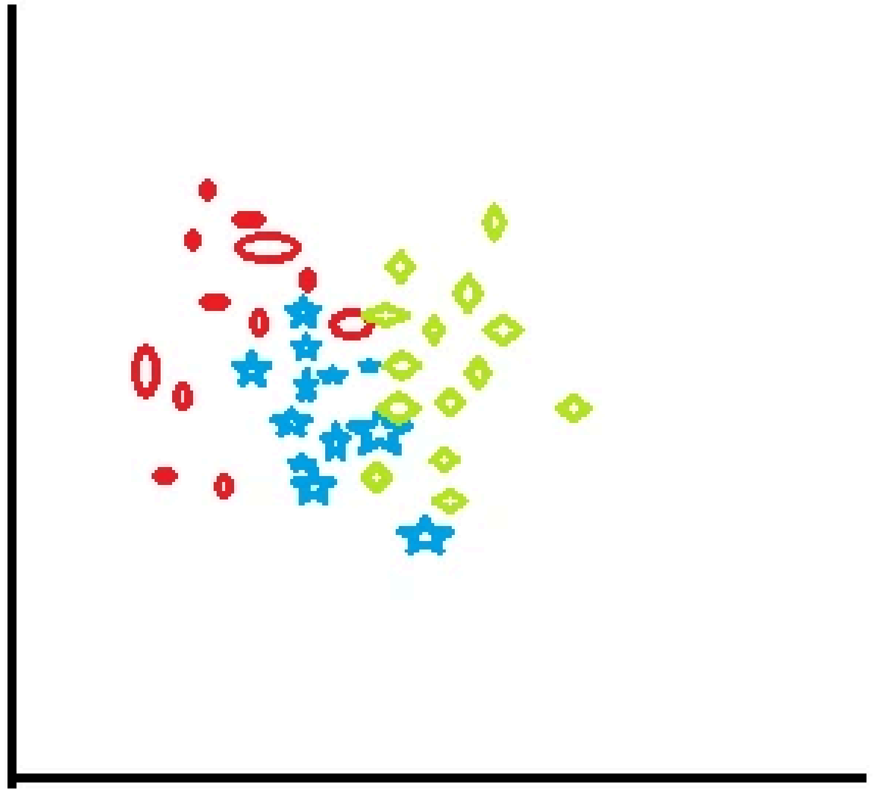


fig 1: before applying k-means clustering

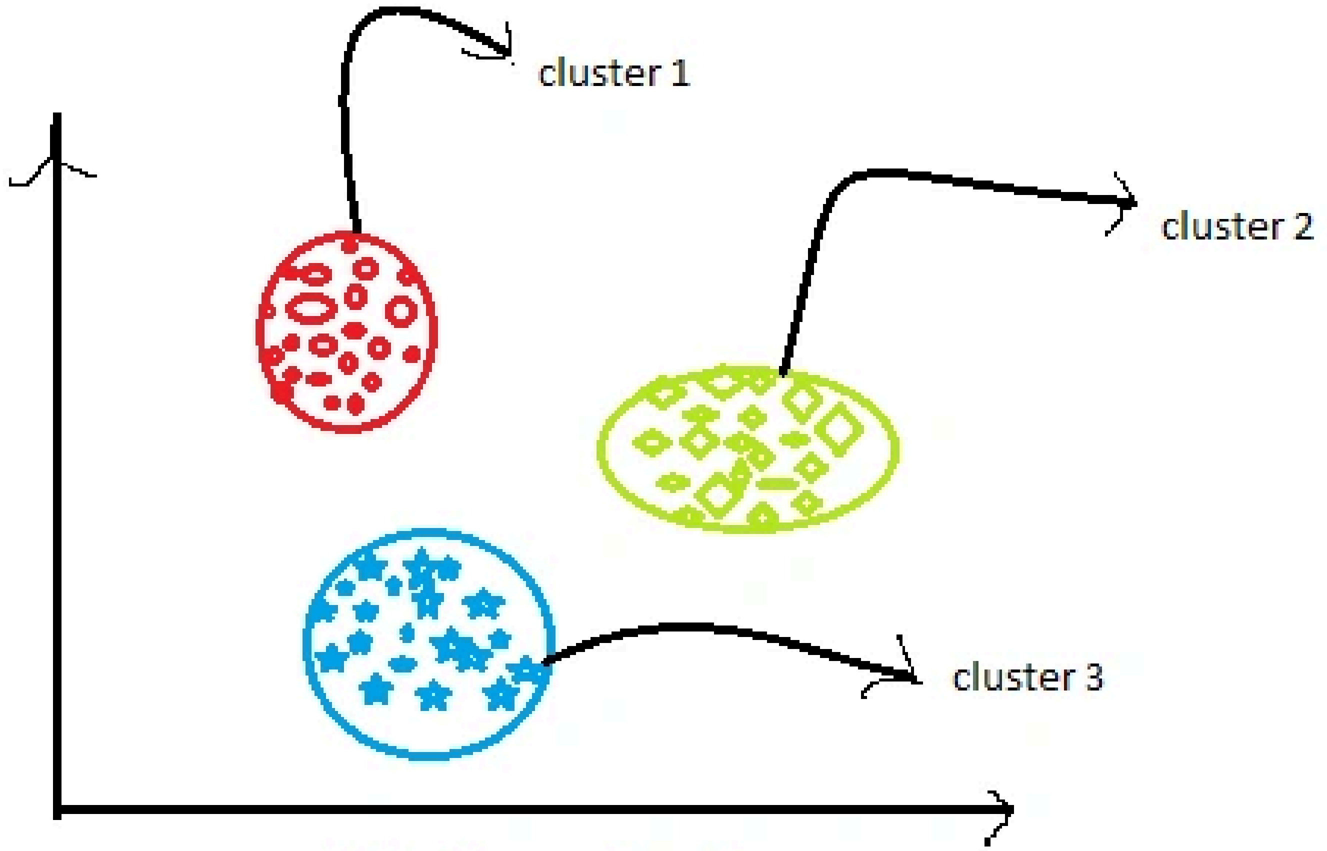


fig 2: After applying K-means clustering

Unsupervised Learning

Question 3: Explain K-Means clustering. What are its main limitations?

Limitations:

o Must choose K in advance.

o Sensitive to outliers and noise.

o Works best with spherical clusters; fails with non-convex clusters.

o Can converge to local minima depending on initialization.

Unsupervised Learning

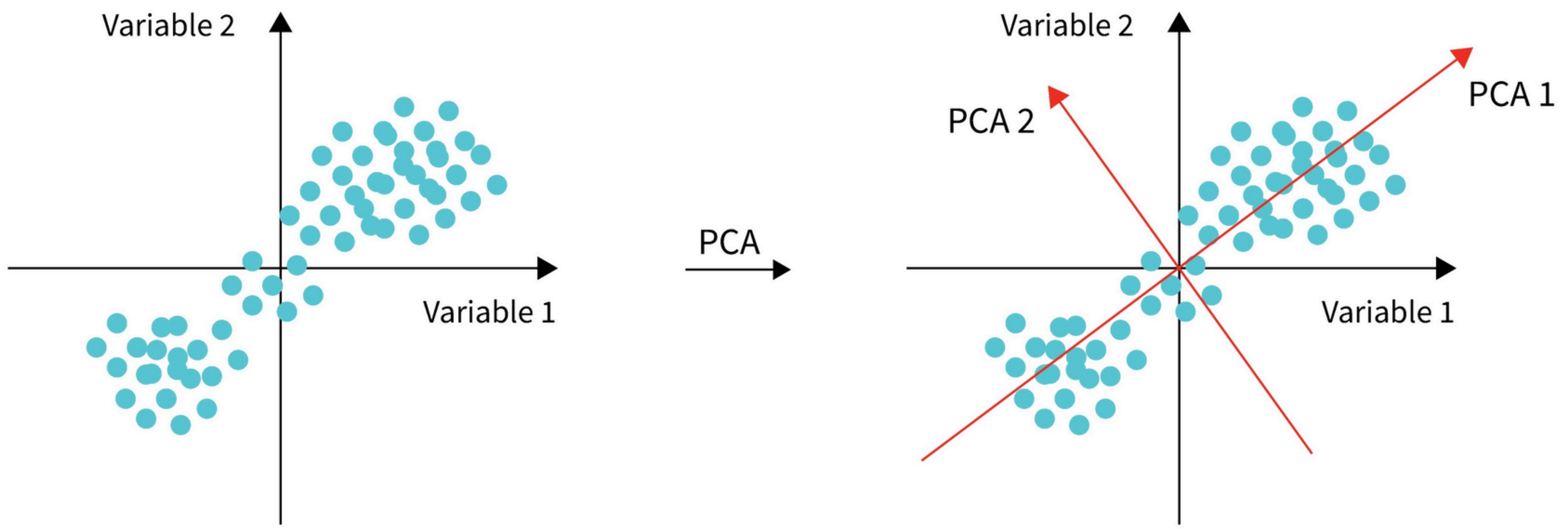
Question 4: How does PCA help in unsupervised learning, and when would you use it?

- Principal Component Analysis (PCA) reduces the dimensionality of data by projecting it onto directions of maximum variance.

Unsupervised Learning

Question 4: How does PCA help in unsupervised learning, and when would you use it?

- Principal Component Analysis (PCA) reduces the dimensionality of data by projecting it onto directions of maximum variance.



Unsupervised Learning

Question 4: How does PCA help in unsupervised learning, and when would you use it?

- Why useful in unsupervised learning:
  - o Reduces computational cost for clustering or visualization.
  - o Removes redundant or correlated features.
  - o Helps detect structure in high-dimensional datasets.
- Example usage:
  - o Reducing a 100-feature dataset to 2–3 components for visualization.
  - o Preprocessing before clustering to improve results.

Unsupervised Learning

Question 5: What is an autoencoder, and how can it be used in unsupervised learning?

Autoencoder: A type of neural network that learns to compress (encode) data into a lower-dimensional space and then reconstruct (decode) it back.

- Applications:

o Dimensionality reduction.

o Anomaly detection: high reconstruction error indicates anomalies.

o Data denoising: remove noise from images or signals.

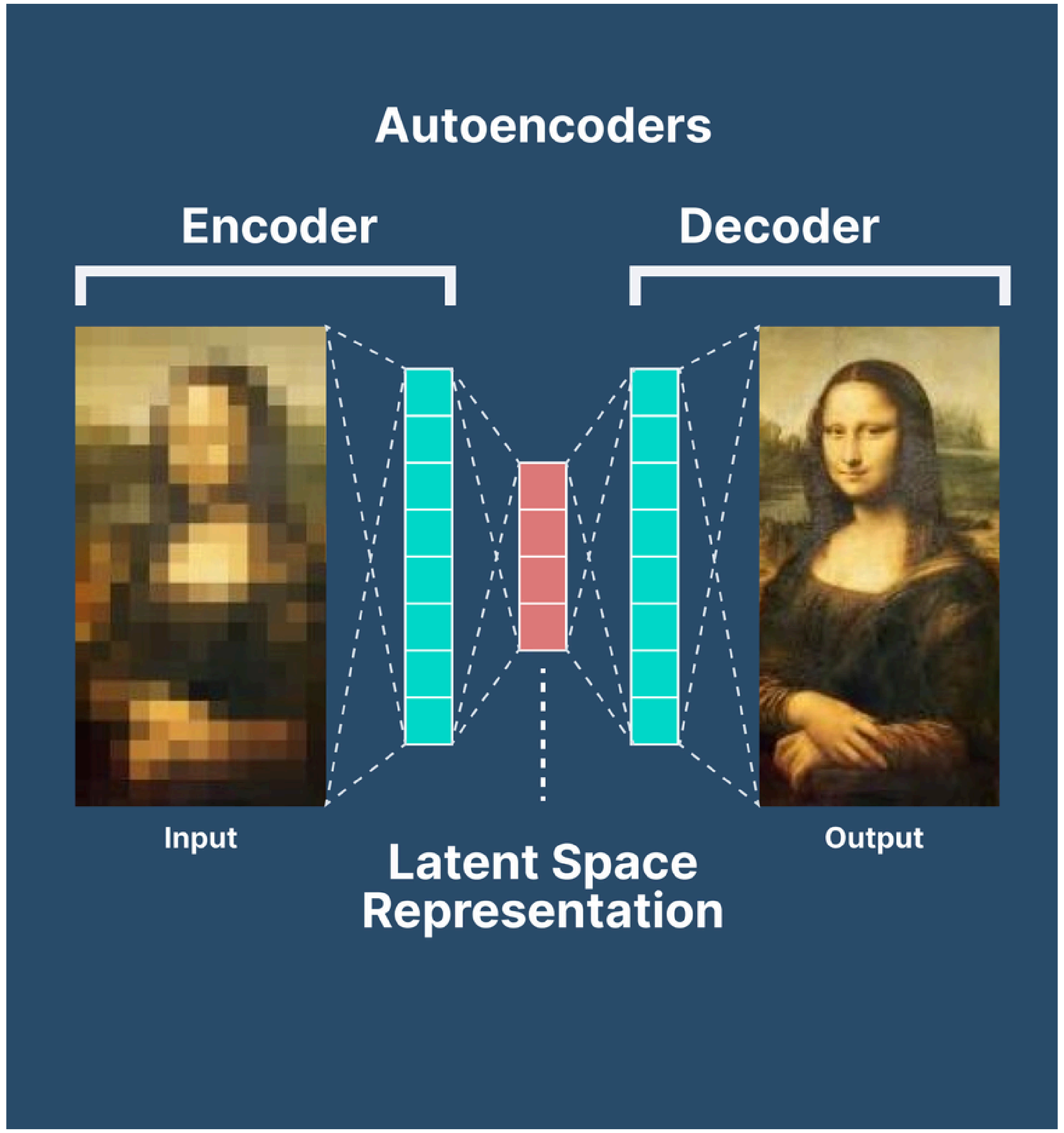
Unsupervised Learning

Question 5: What is an autoencoder, and how can it be used in unsupervised learning?

• Architecture:

o Encoder: compresses input to latent representation.

o Decoder: reconstructs input from latent representation.



Unsupervised Learning

Question 6: Give an example of a real-world problem where unsupervised learning is preferred over supervised learning.

- Customer segmentation in e-commerce:
  - No predefined labels for types of customers.
  - Clustering can reveal natural groups (e.g., bargain hunters, premium buyers) to target marketing campaigns.
- Other examples:
  - Detecting unusual network activity (anomaly detection).
  - Compressing high-dimensional image data (dimensionality reduction).

Unsupervised Learning

Question 7: What are the main components of an autoencoder, and how do they work?

- An autoencoder consists of three main parts:

1. Encoder:

o Maps the input data $x$to a lower-dimensional latent representation z.

o Learns the most important features of the data.

Unsupervised Learning

Question 7: What are the main components of an autoencoder, and how do they work?

2. Latent Space (Bottleneck):

o The compressed representation z that contains essential information about the input.

o Forces the network to capture meaningful patterns rather than memorizing data.

Unsupervised Learning

Question 7: What are the main components of an autoencoder, and how do they work?

2. Latent Space (Bottleneck):

o The compressed representation z that contains essential information about the input.

o Forces the network to capture meaningful patterns rather than memorizing data.

Unsupervised Learning

Question 7: What are the main components of an autoencoder, and how do they work?

3. Decoder:

o Reconstructs the original input from the latent representation z.

o Produces output $\hat{x}$ that approximates the input x.

• Goal: Minimize the difference between x and $\hat{x}$, i.e., the reconstruction error.

Unsupervised Learning

Question 8: What kind of loss function is typically used to train an autoencoder, and why?

Common choices: 1. Mean Squared Error (MSE):

$$L = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

- Penalizes differences, penalizes large differences more heavily.
- Suitable for continuous data.

Unsupervised Learning

Question 8: What kind of loss function is typically used to train an autoencoder, and why?

2. Binary Cross-Entropy

$$\text{BCE Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[x_i \cdot \log(\hat{x}_i) + (1 - x_i) \cdot \log(1 - \hat{x}_i)\right]$$

Unsupervised Learning

Question 8: What kind of loss function is typically used to train an autoencoder, and why?

$$\text{BCE Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[x_i \cdot \log(\hat{x}_i) + (1 - x_i) \cdot \log(1 - \hat{x}_i)\right]$$

- Used when inputs are binary or normalized between 0 and 1.
- If a pixel is 1, the AE is penalized if $\hat{x}i$ is far from 1.
- If a pixel is 0, the AE is penalized if $\hat{x}i$ is far from 0.
- This is different from MSE, which penalizes differences in absolute value rather than probability. BCE focuses on probabilistic reconstruction, which often produces sharper outputs for image data.

• Reason: Minimizing this loss ensures the encoder-decoder pair learns a latent representation that preserves the essential structure of the data.

Unsupervised Learning

Question 9: How do we use autoencoders in practice?

1. Dimensionality Reduction / Feature Extraction:

o Use the encoder to compress high-dimensional data into a smaller, informative latent representation.

o The latent features can then be used for downstream tasks like clustering, classification, or visualization.

2. Data Denoising:

o Train an autoencoder to reconstruct clean data from noisy inputs.

o The model learns to ignore noise and preserve the underlying signal.

Unsupervised Learning

Question 9: How do we use autoencoders in practice?

3. Anomaly Detection:

o Train the autoencoder on normal data.

o High reconstruction error indicates unusual or anomalous inputs.

4. Data Compression:

o The encoder output can serve as a compressed version of the input, useful for storage or transmission.

Unsupervised Learning

Question 9: How do we use autoencoders in practice?

5. Pretraining for Neural Networks:

o Use the encoder as a feature extractor to initialize weights for other models, improving learning on small labeled datasets.

• Workflow example:

1. Input data → encoder → latent representation
2. Latent representation → decoder → reconstructed output
3. Use latent representation for tasks such as feature extraction, dimensionality reduction, clustering, or check reconstruction error for anomaly detection.