International Journal of
Digital Communication and Analog Signals

ISSN: 2455-0329
Volume 11, Issue 1, 2025
January–June
DOI (Journal):10.37628/IJDCAS

https://journalspub.com/journal/IJDCAS/

Review IJDCAS

# Unsupervised Machine Learning: Revealing Hidden Data Patterns

Vansh Mudgal[1,*], Aaditya Sharma[2]

***Abstract***

*Machine learning helps computers learn from data and make smart decisions. There are two main ways they do this: supervised and unsupervised learning. Supervised learning is like teaching a child with flashcards – you show the computer examples with the right answers, and it learns to recognize patterns. This helps it make predictions, like guessing whether an email is spam or estimating house prices. Unsupervised learning, on the other hand, is like giving a child a puzzle without a picture guide. The computer must find patterns on its own. For example, K-means clustering groups similar items together, like organizing songs by genre without being told what each genre is. Other methods, like anomaly detection and PCA, help spot unusual data points or simplify large amounts of information. There's also collaborative filtering, which is how streaming services suggest movies based on what you've watched. And then there's reinforcement learning, where a computer learns by trial and error – like a robot figuring out how to walk by trying different movements and adjusting based on feedback. These methods are used everywhere, from predicting trends to making apps more user-friendly. They're shaping the future of technology and making everyday life more efficient.*

**Keywords:** Supervised learning, unsupervised learning, K-means clustering, anomaly detection, PCA algorithm, collaborative filtering, reinforcement learning

## INTRODUCTION

Machine learning operates much like teaching a child to recognize cats and dogs. By showing images and providing correct labels, the child gradually learns to identify them independently. Similarly, machine learning enables computers to detect patterns, make predictions, and improve performance over time without requiring explicit programming for every task.

This technology is deeply integrated into everyday life, often without users realizing it. It powers personalized recommendations on streaming platforms, filters spam emails, enhances voice assistants, and enables autonomous vehicles. Machine learning occurs through different approaches: supervised learning, which relies on labeled data; unsupervised learning, which uncovers hidden patterns; and reinforcement learning, which improves performance through trial and error, much like a video game character refining skills after multiple attempts [1–5].

***Author for Correspondence**
Vansh Mudgal
E-mail: mudgal.vanshu@gmail.com

[1]Student, Department of Information Technology, Inderprastha Engineering College, Sahibabad, Ghaziabad, Uttar Pradesh, India
[2]Student, Department of Electronics and Communication Engineering, Inderprastha Engineering College, Sahibabad, Ghaziabad, Uttar Pradesh, India

Received Date: March 11, 2025
Accepted Date: March 20, 2025
Published Date: March 29, 2025

**Citation:** Vansh Mudgal, Aaditya Sharma. Unsupervised Machine Learning: Revealing Hidden Data Patterns. International Journal of Digital Communication and Analog Signals. 2025; 11(1): 38–47p.

## A SUPERVISED LEARNING ENVIRONMENT

Supervised learning occupies a central role in the field of machine learning, empowering algorithms to effectively discover the hidden relationships between input data and

corresponding outputs. This learning paradigm relies on a labeled dataset, where each data point is paired with its correct outcome, acting as a guidepost for the algorithm's development. By analyzing these paired examples, the algorithm learns to recognize patterns and builds a model capable of predicting or classifying new, unseen data [6, 7].

The essence of supervised learning lies in the iterative process of feeding the algorithm labeled data and allowing it to refine its understanding of the underlying patterns. Through this process, the algorithm gradually develops a mapping function that translates new input data into accurate and reliable output predictions. This ability to generalize from training data to unseen examples makes supervised learning invaluable across a wide range of disciplines, including,

- *Image Recognition:* Recognizing and classifying objects within images.
- *Natural Language Processing:* Extracting meaning and insights from human language.
- *Speech Recognition:* Converting spoken language into text.
- *Medical Diagnosis:* Predicting medical conditions based on patient data.
- *Financial Forecasting:* Predicting future trends in financial markets.

The success of supervised learning hinges on the quality and quantity of the labeled data. An insufficient or poorly labeled dataset can lead to inaccurate predictions and hinder the algorithm's ability to generalize effectively. Therefore, careful data preparation and labeling are crucial for building robust and reliable supervised learning models.

**Formal Representation**

Supervised learning can be formally represented as a dataset containing tuples of input data (x) and corresponding output labels

(y): $[(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)]$

where,

- x represents the input data, typically a vector of features or attributes.
- y represents the output label, which can be a discrete value (classification) or a continuous value (regression).

By analyzing these labeled data points, the supervised learning algorithm learns to construct a model capable of accurately predicting the output label y for a new, unseen input data x.

**Learning Environment Unsupervised**

Unsupervised learning constitutes a powerful branch of machine learning, capable of extracting meaningful insights from data devoid of explicit labels. Unlike its supervised counterpart, where algorithms are guided by pre-defined outputs, unsupervised learning empowers algorithms to discover inherent patterns, relationships, and structures within the data autonomously. This self-discovery process allows algorithms to identify underlying trends and organize data into meaningful clusters without the need for human intervention [8–10].

The true power of unsupervised learning lies in its ability to handle large datasets where labeling every data point would be impractical or prohibitively expensive. This makes it ideal for scenarios involving massive datasets in fields like,

- *Exploratory Data Dnalysis:* Gaining initial insights into the characteristics and underlying trends within uncharted data.
- *Customer Segmentation:* Grouping customers based on shared behaviors or preferences for targeted marketing campaigns.
- *Image Segmentation:* Identifying and separating objects within complex images for object recognition tasks.

- *Anomaly Detection:* Detecting unusual or outlier data points that deviate from expected patterns.
- *Dimensionality Reduction:* Reducing the number of features in a dataset while preserving essential information.

Unsupervised learning algorithms typically rely on unlabeled datasets, represented as,

$[(x_1), (x_2), ..., (x_n)]$

where,

- $x_i$ represents an individual data point, typically a vector of features or attributes.

By analyzing these unlabeled data points, algorithms, like clustering and dimensionality reduction, identify similarities and relationships between data points, ultimately grouping them into meaningful clusters or reducing their dimensionality without compromising essential information. This ability to learn from unlabeled data renders unsupervised learning a valuable tool for exploring, understanding, and manipulating vast data landscapes.

**K-Means Clustering**

K-means clustering is a widely used unsupervised learning technique that excels at identifying inherent structures within unlabeled datasets. It relies solely on the intrinsic similarities between data points, allowing it to discover meaningful patterns and clusters without prior knowledge or explicit labeling.

**The K-Means Algorithm: Finding Unity and Diversity**

The K-means algorithm operates in an iterative fashion, starting with an unlabeled data collection and a pre-defined number of clusters (k). It then iteratively performs the following steps,

- *Centroid Initialization:* K initial centroids, representing the hypothetical centres of the clusters, are randomly chosen within the data space.
- *Point Assignment:* Each data point is assigned to the closest centroid based on a distance measure, typically Euclidean distance.
- *Centroid Re-computation:* The centroids are recomputed by calculating the average of all data points assigned to each individual cluster.
- *Cluster Reassignment:* Data points are reassigned to the closest centroid after the centroids have been updated.
- *Iteration:* Steps 2–4 are repeated until the centroids converge and no further changes occur in the cluster assignments.

This iterative process ensures that data points within a cluster are highly like each other, while points belonging to different clusters exhibit significant variations.

**Real-World Applications for K-Means Clustering**

K-means clustering finds diverse applications across various domains, including

- *Customer Segmentation:* Grouping customers with similar buying patterns for target marketing campaigns.
- *Image Segmentation:* Identifying and isolating objects within images for image recognition tasks.
- *Gene Expression Analysis:* Clustering genes based on their expression profiles to discover co-regulated genes.
- *Anomaly Detection:* Identifying unusual data points that deviate from expected patterns within a cluster.
- *Document Clustering:* Grouping documents with similar topics for efficient information retrieval.

The versatility and ease of implementation of the K-means algorithm make it a powerful tool for exploring unlabeled data and uncovering hidden structures within complex datasets.

**Visual Representation of K-Means Clustering**

Imagine a graph with diverse data points scattered across it. The K-means algorithm initially places colored "flags" (centroids) at random locations on the graph. Each data point is then assigned to the flag with the closest color, forming initial clusters. The colors of the flags are then recalculated based on the average color of the points assigned to each one. This process continues iteratively, with points potentially switching allegiance between clusters until the flags and points stabilize. Ultimately, the data points are grouped into distinct clusters based on their inherent similarities, forming a clear visual representation of the underlying structure within the data.

The centroid of each cluster, or the average of all the data points inside the cluster, is then determined via the K-means procedure.

**Process**

Figure 1 shows a clustering process in which data points are grouped according to their similarities. Individual data points are represented by the yellow dots, while the centroids of two respective clusters are denoted by the red and blue crosses. The arrows show how the centroids move as the clustering algorithm works to optimize cluster positions. The expression $\{x^{(1)}, x^{(2)}, x^{(3)}, ..., x^{(30)}\}$ indicates a dataset containing several feature vectors. Cluster centroids (marked in blue) indicate the central points where data points are clustered. This visualization is often employed in K-means clustering and other unsupervised learning methods to depict the formation and development of clusters in multidimensional spaces.

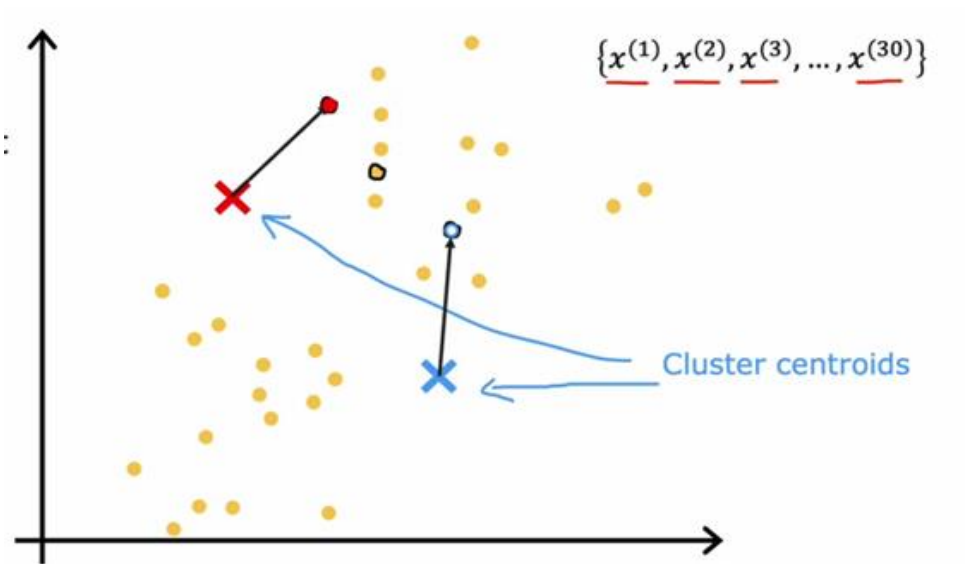
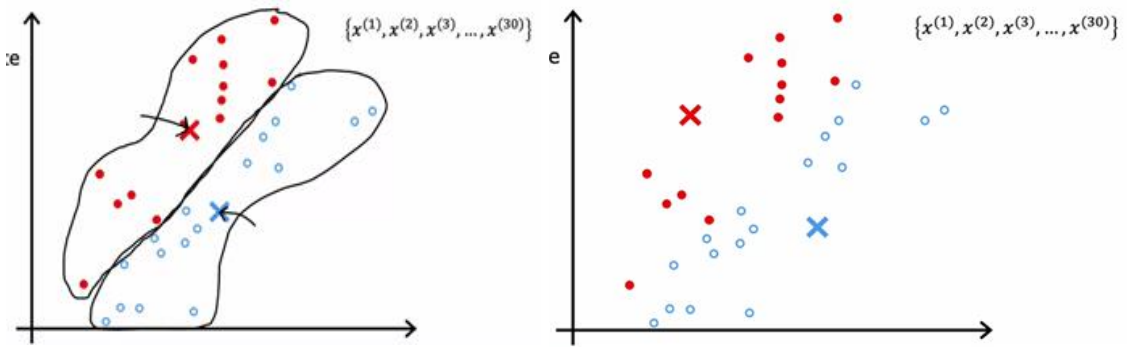



**Figure 1.** Block diagram depicting centroids of the respective clusters.

**Recompute the Centroids**

Figure 1 illustration shows a comparison of the raw data distribution with classification decision boundaries. In 2(a), two different classes are indicated by red and blue dots, while larger crosses mark important data points. The dataset comprises feature vectors represented as $\{x^{(0)}, x^{(1)}, x^{(2)}, ..., x^{(208)}\}$ along the x-axis. In 2(b), decision boundaries (depicted as black curves) are presented, demonstrating how a classification model distinguishes between the two classes. Advanced classification techniques, such as kernel-based SVM or deep learning are suggested by the non-linear shape of the boundaries. This Figure 2 emphasizes the conversion of unrefined data into a structured decision space for efficient classification.




(a) Data Distribution. (b)Decision Boundary in a Classification Task.

**Figure 2(a, b).** In the above diagram the respective centroids are positioned according to clusters.

**Elbow Method**

The elbow method is used to choose the value of K or the number of clusters. The elbow method involves plotting the within-cluster sum of squares against the number of clusters, K. The point at which the slope of the curve begins to flatten is the elbow point, which indicates the optimal number of clusters. In this method we plot a graph between Cost function (J) and no. of Clusters (K).

The graph (Figure 3) decreases as the number of clusters increases. We choose the value of k where the graph decreases less rapidly. For example, the graph decreases less slowly after k = 3 so the value of k is 3.

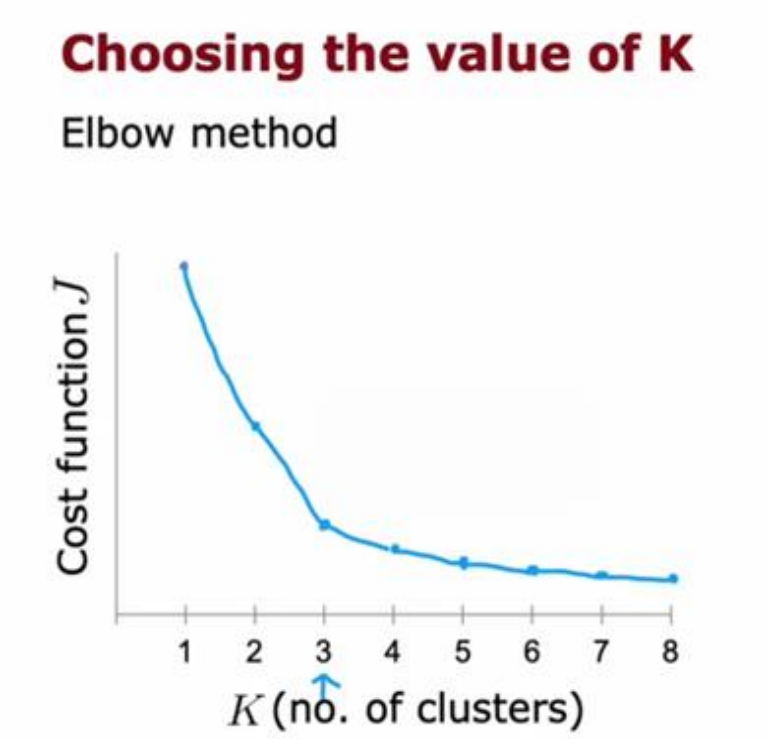



**Figure 3.** Value of cost function decreases with increase in value of k at k=3 the curve begins to decrease less gradually.

**Drawback**

The drawback of this method is that when there can be cases where the graph is decreasing normally without any significant difference in such cases, the decreasing trend may not be statistically significant enough to conclude that there is a trend in the data. Additionally, this method is not suited for use with

data that contains outliers, as these can skew the results and make it difficult to detect a trend. For example, in the graph below (Figure 4), it's difficult to find value for K.

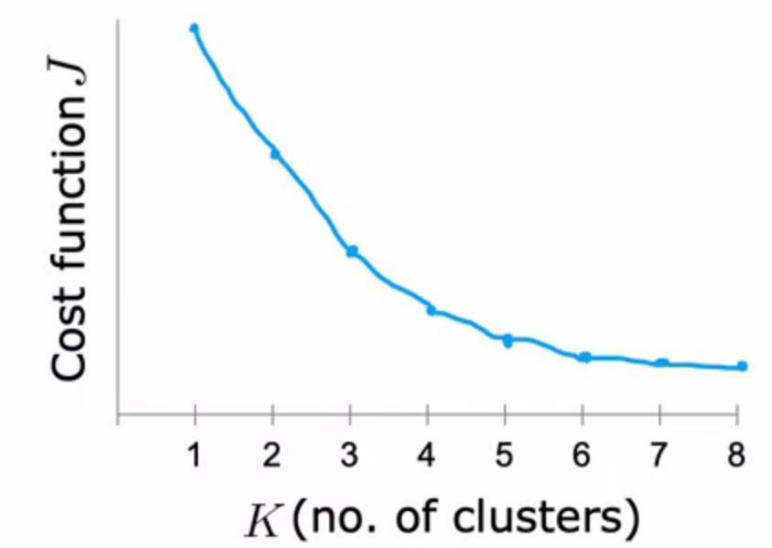



**Figure 4.** Cost function decreases as no. of clusters increase.

**Anamoly Detection**

In machine learning, anomaly detection is the process of finding patterns or examples within a dataset that drastically depart from the norm. These observations – also referred to as anomalies or outliers – are uncommon, uncommon, or deviate from typical patterns of activity.

Making the distinction between normal and abnormal data points is the aim of anomaly detection. This is especially helpful in a variety of applications where anomalies may point to important problems or insightful information, such as fraud detection, network security, industrial quality control, and healthcare monitoring.

Let us understand this topic with the help of an example of an Aircraft Engine. Let an aircraft engine have 2 features.

- $X_1$ = heat generated.
- $X_2$ = vibration intensity.

Let us take a dataset $\{x_1, x_2, x_m\}$ called M engines.

The dataset is made up of aircraft engines that were manufactured previously. An engine called X test is introduced, and its state – normal or defective – must be assessed. A graph (Figure 5) is created to evaluate this, with $X_2$ (Vibration Intensity) represented on one axis and $X_1$ (Heat Generated) on the other. Every data point in the dataset that represents the engines (M engines) is plotted. If the X test falls within these data points, it is categorized as a normal engine. If it is identified as an outlier, however, it is deemed defective.

The probability of $x_{test}$ lying within the dataset depends upon value of ε (Figure 6).

- If $p(x_{test}) < \varepsilon$ then $x_{test}$ is an anamoly as it lies outside the normal distribution of dataset.
- If $p(x_{test}) > \varepsilon$ then $x_{test}$ is not an anamoly as it lies within the normal distribution of dataset.

**PCA ALGORITHM**

Initially the data is plotted using two axis $x_1$ and $x_2$, but supposedly the data is plotted using only one axis that is z axis (Figure7).

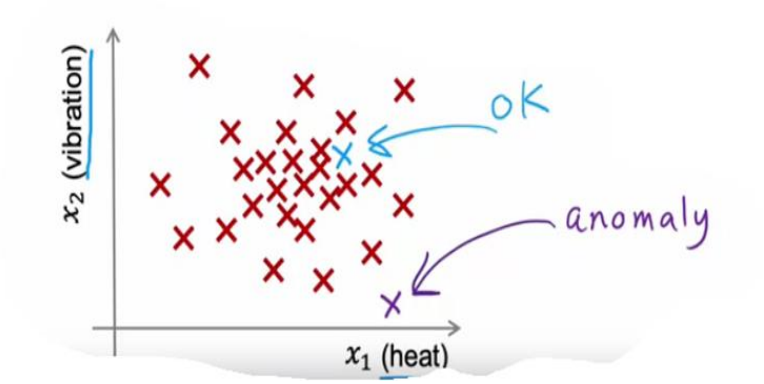



**Figure 5.** Block diagram of for anamoly detection.

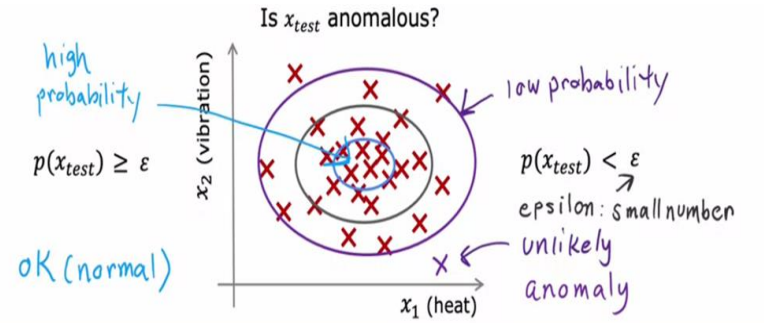



**Figure 6.** Block diagram representing probability of $x_{test}$ lying within dataset.

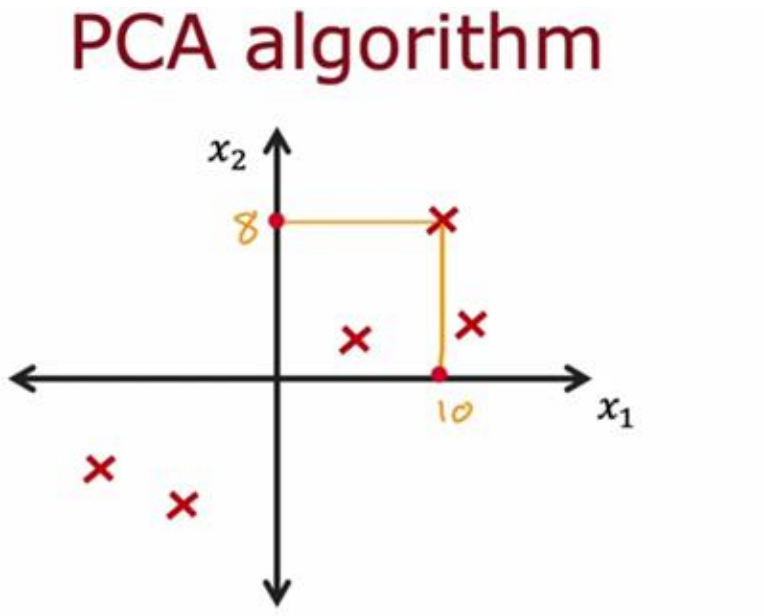



**Figure 7.** In the above diagram, data is plotted using two axis x and y.

This Figure 8 shows different points spread along a horizontal line (z-axis), with a black dot marking the starting point, called the origin. The red crosses represent data points, and the vertical lines indicate how far each point is from the z-axis. The orange arrows show the distances between different points, helping to understand how they are positioned relative to the origin.

This Figure 9 illustrates the concept of Principal Component Analysis (PCA) in an unsupervised machine learning context. The red crosses represent data points distributed in a multi-dimensional

space. The principal component, shown as a blue arrow, indicates the direction of maximum variance in the dataset, helping to reduce dimensionality while retaining essential information.

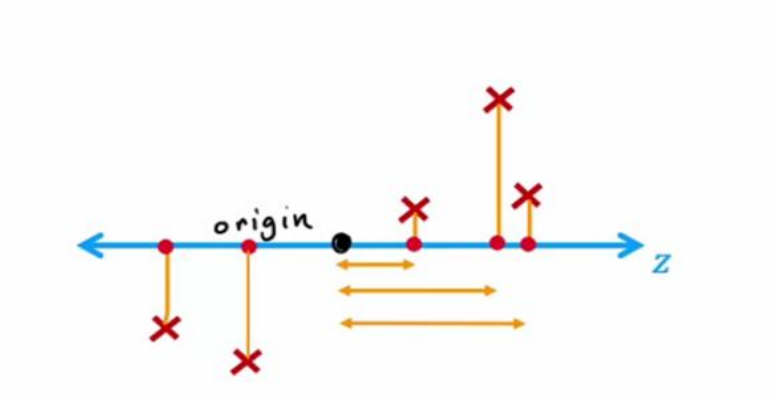



**Figure 8.** In the above diagram, the data is plotted using only one axis.

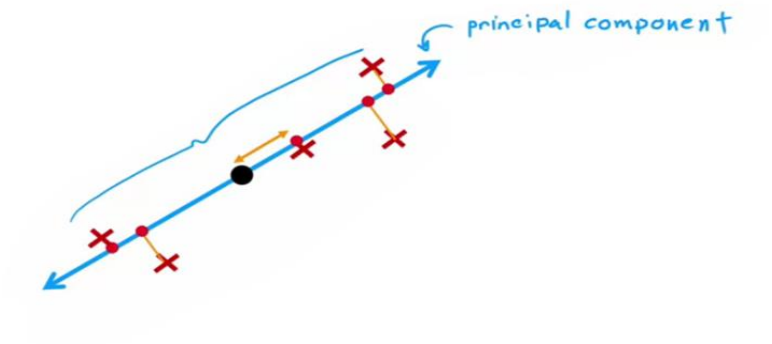



**Figure 9.** Concept of Principal Component Analysis (PCA).

In this example (Figure 10), points have a large variance and are spread apart. The line on which these points are spread is called principal component. Principal components affect the projection.

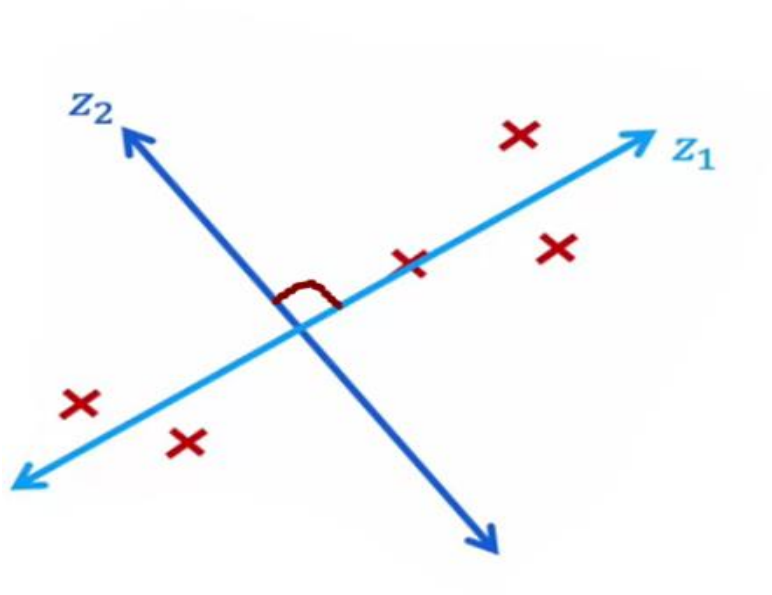



**Figure 10.** Two principal components that are perpendicular to each other.

There can be two principal components, but they must be perpendicular to each other. Similar, we have 3 principal components perpendicular to each other. There can be no principal components, but they must be perpendicular to each other see in Figure 11.

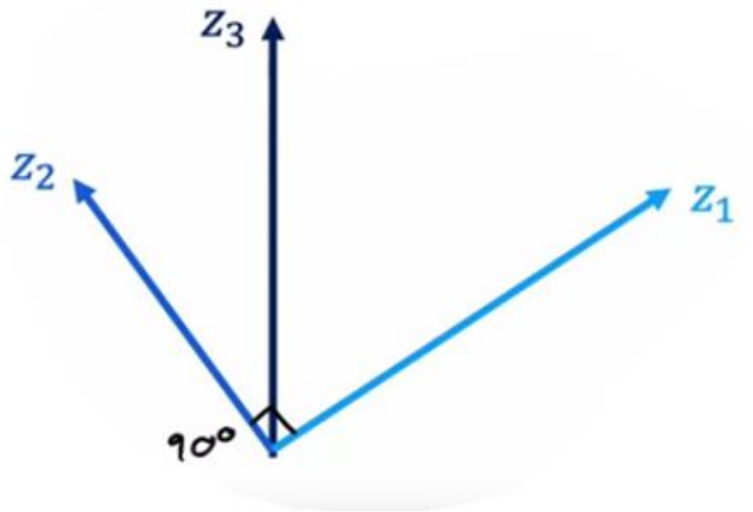



**Figure 11.** Figure representing 3 principal components perpendicular to each other.

**Collaborative Filtering**

Recommendation systems frequently employ the method of collaborative filtering to infer user preferences by utilizing the actions and preferences of a group of users. The theory is that if two users, A and B, have similar tastes for some products and A likes something that B hasn't tried yet, then B could like that same item as well.

Two primary categories of collaborative filtering exist as under,

**Collaborative Filtering Based on Users**

Using the preferences of users who are like the target user, this method suggests goods or things.

The algorithm finds users who have the target user's tastes and suggests products that those users have enjoyed or used.

Metrics, like Pearson correlation or cosine similarity, are frequently used to gauge how similar two users are to one another.

**Item-Based Collaborative Filtering**

Using this method, the system suggests products based on what a user has enjoyed or engaged with in the past.

It finds products that resemble the ones the user has expressed interest in and suggests them.

A common method for calculating item similarity is to use a methodology like Pearson correlation or cosine similarity.

**Reinforcement Learning**

A kind of machine learning called reinforcement learning (RL) teaches an agent to make decisions by interacting with its surroundings. The main concept is that the agent should learn from mistakes and experimentation to determine what actions produce the best results under various conditions.

- *Agent:* The learner or decision-maker is the agent. It is the thing that acts in a setting to accomplish a purpose.
- *Environment:* The external system that the agent interacts with is referred to as the environment. It could be a virtual environment (like a computer program playing a game) or an actual one (like a robot wandering around a room).
- *State:* The environment is in a particular state at any given time. This is an illustration of the state or arrangement now.
- *Action:* The agent can make choices or motions that have an impact on the surroundings. The present condition determines the range of feasible actions.
- *Reward*: The agent is given a reward for acting in a particular state. This is a numerical value that represents the agent's current performance. Usually, the agent's objective is to maximize the reward's cumulative value over time.

## CONCLUSIONS

Machine learning helps computers learn from data and make smart choices. It works in two main ways: supervised learning, where the computer learns from labeled examples (like a student learning with answer keys), and unsupervised learning, where it finds patterns on its own (like sorting puzzle pieces without a reference image). Techniques, like K-means clustering group similar items, anomaly detection spots unusual data, and collaborative filtering, suggest things based on past choices (like movie recommendations). This article covers supervised and unsupervised learning, reinforcement learning, and collaborative filtering, highlighting techniques like K-means clustering and PCA. These methods have diverse applications, from classification and anomaly detection to recommendation systems. As advancements continue, machine learning will play an increasingly vital role in automation, efficiency, and decision-making across various industries.

## REFERENCES

1. Putri DCG, Leu JS, Seda P. Design of an unsupervised machine learning-based movie recommender system. Symmetry. 2020 Jan 21;12(2):185.
2. Naeem S, Ali A, Anam S, Ahmed MM. An unsupervised machine learning algorithm: Comprehensive review. Int J Comput Digit Syst. 2023 Mar 2.
3. Watson DS. On the philosophy of unsupervised learning. Philos Technol. 2023 Jun;36(2):28.
4. Maier MI, Czibula G, Oneț-Marian ZE. Towards using unsupervised learning for comparing traditional and synchronous online learning in assessing students' academic performance. Mathematics. 2021 Nov 11;9(22):2870.
5. Sarker IH. Machine learning: Algorithms, real-world applications and research directions. SN Comput Sci. 2021 May;2(3):160.
6. Wang J, Biljecki F. Unsupervised machine learning in urban studies: A systematic review of applications. Cities. 2022 Oct 1;129:103925.
7. Murali MV, Vishnu TG, Victor N. A collaborative filtering-based recommender system for suggesting new trends in any domain of research. In: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS); 2019 Mar 15. p. 550–3. IEEE.
8. Maćkiewicz A, Ratajczak W. Principal components analysis (PCA). Comput Geosci. 1993 Mar 1;19(3):303–42.
9. Patcha A, Park JM. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput Networks. 2007 Aug 22;51(12):3448–70.
10. Singh S, Gill NS. Analysis and study of K-means clustering algorithm. Int J Eng Res Technol. 2013 Jul;2(7):2546–51.