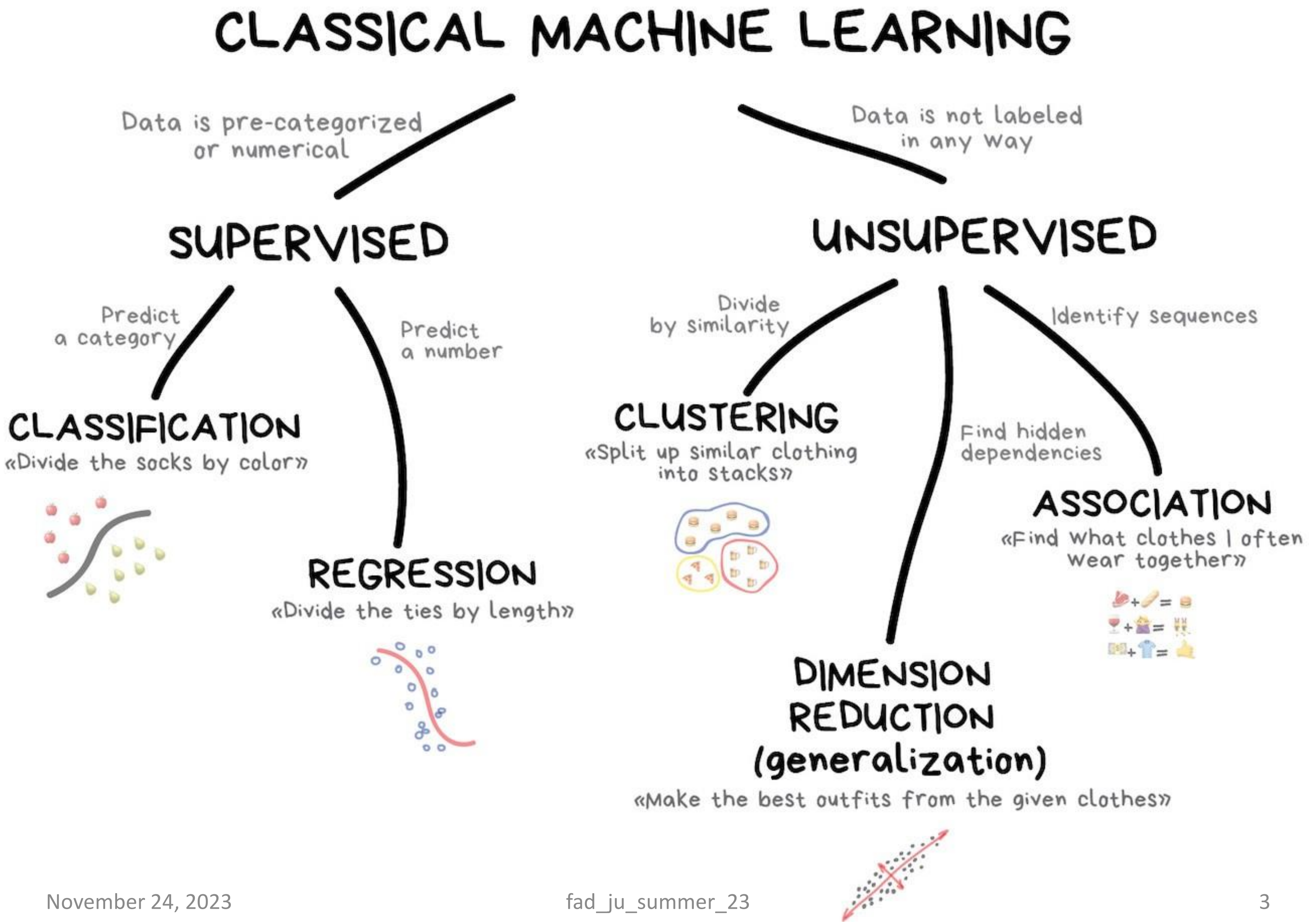
Introduction to Data Science with Python

WMASDS04

Week 10: Unsupervised Learning

Outlines:

- Unsupervised Learning
- Clustering
- Similarity and Distances
- Metrics for Evaluating Clustering Quality
- Rand Index, Homogeneity, Completeness and V-measure Scores
- Silhouette Score
- Optimum Number of Clustering
- Elbow Method
- Techniques: K-means Clustering, Hierarchical Clustering



CLASSICAL MACHINE LEARNING

Data is pre-categorized or numerical

Data is not labeled in any way

SUPERVISED

UNSUPERVISED

Predict a category

Predict a number

CLASSIFICATION

«Divide the socks by color»

REGRESSION

«Divide the ties by length»

Divide by similarity

Find hidden dependencies

Identify sequences

CLUSTERING

«Split up similar clothing into stacks»

DIMENSION REDUCTION (generalization)

«Make the best outfits from the given clothes»

ASSOCIATION

«Find what clothes I often wear together»

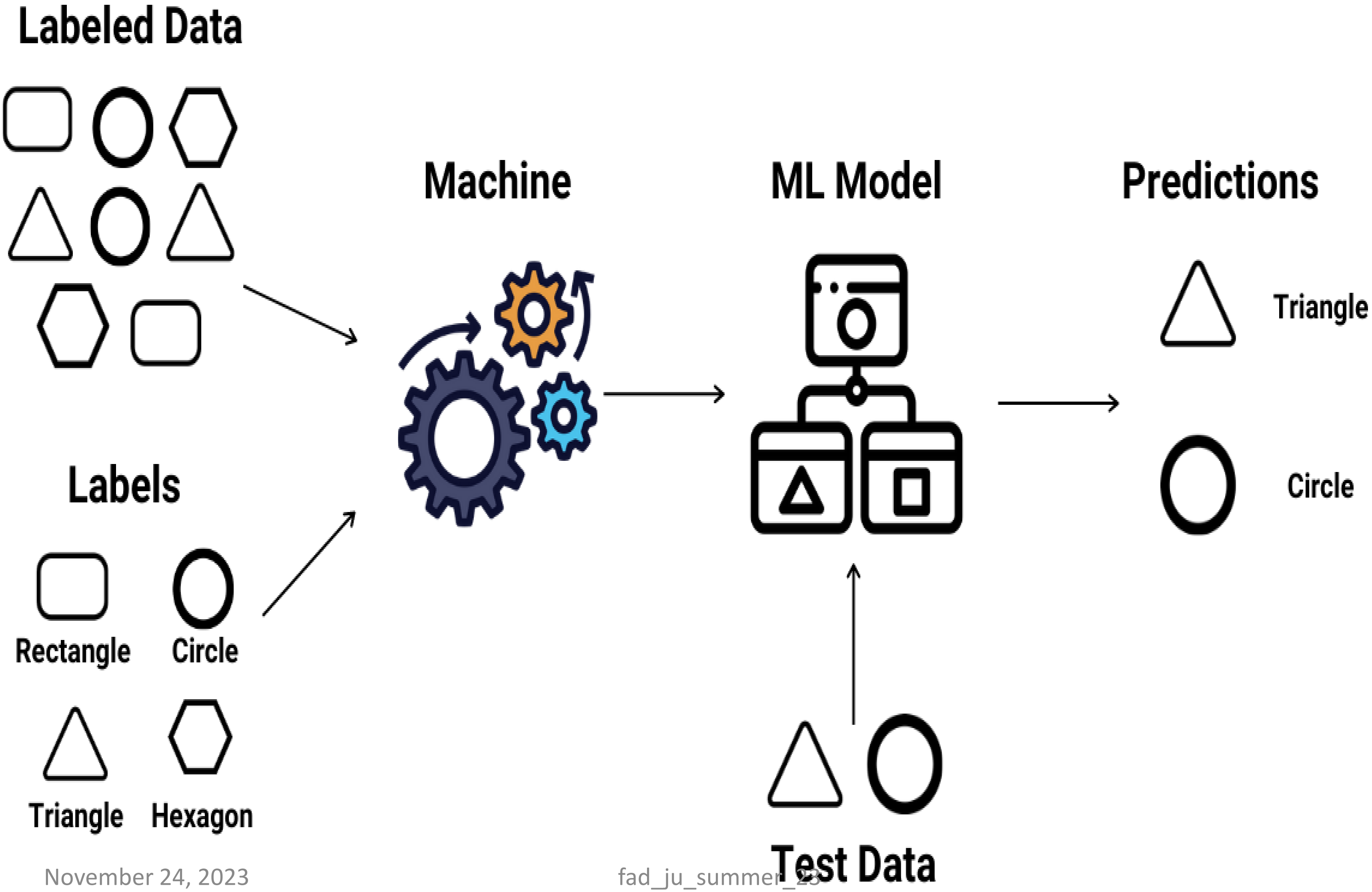
Machine Learning

- Machine learning is a subfield of artificial intelligence (AI) which provides machines the ability to learn automatically and improve from experience without being explicitly programmed.

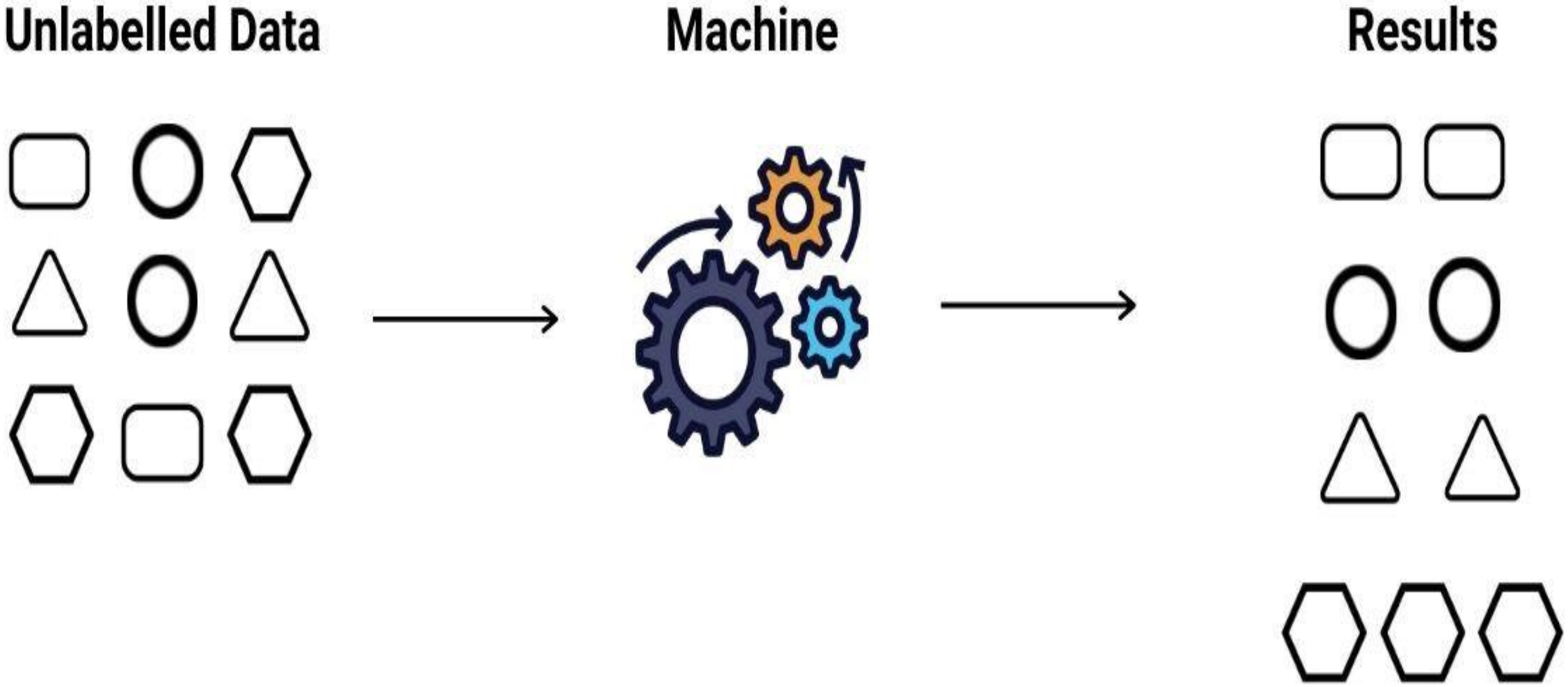
Types of Machine Learning

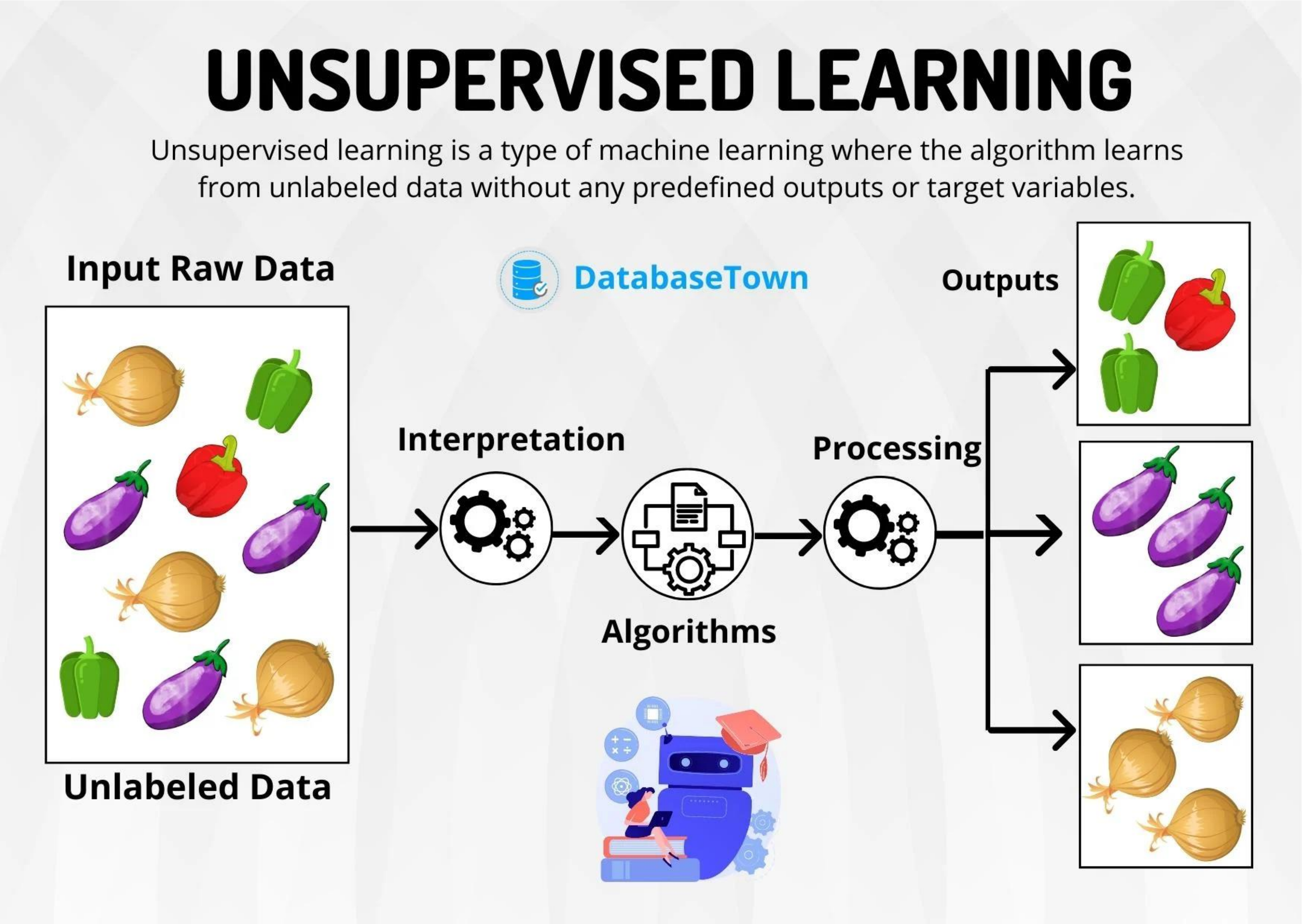
- Supervised Learning
  - Regression Problems
  - Classification Problems
- Unsupervised Learning
  - Clustering Problem
  - Association Problems
  - Dimension Reduction
  - Anomaly detection problem
- Reinforcement Learning

Supervised Learning



Unsupervised Learning

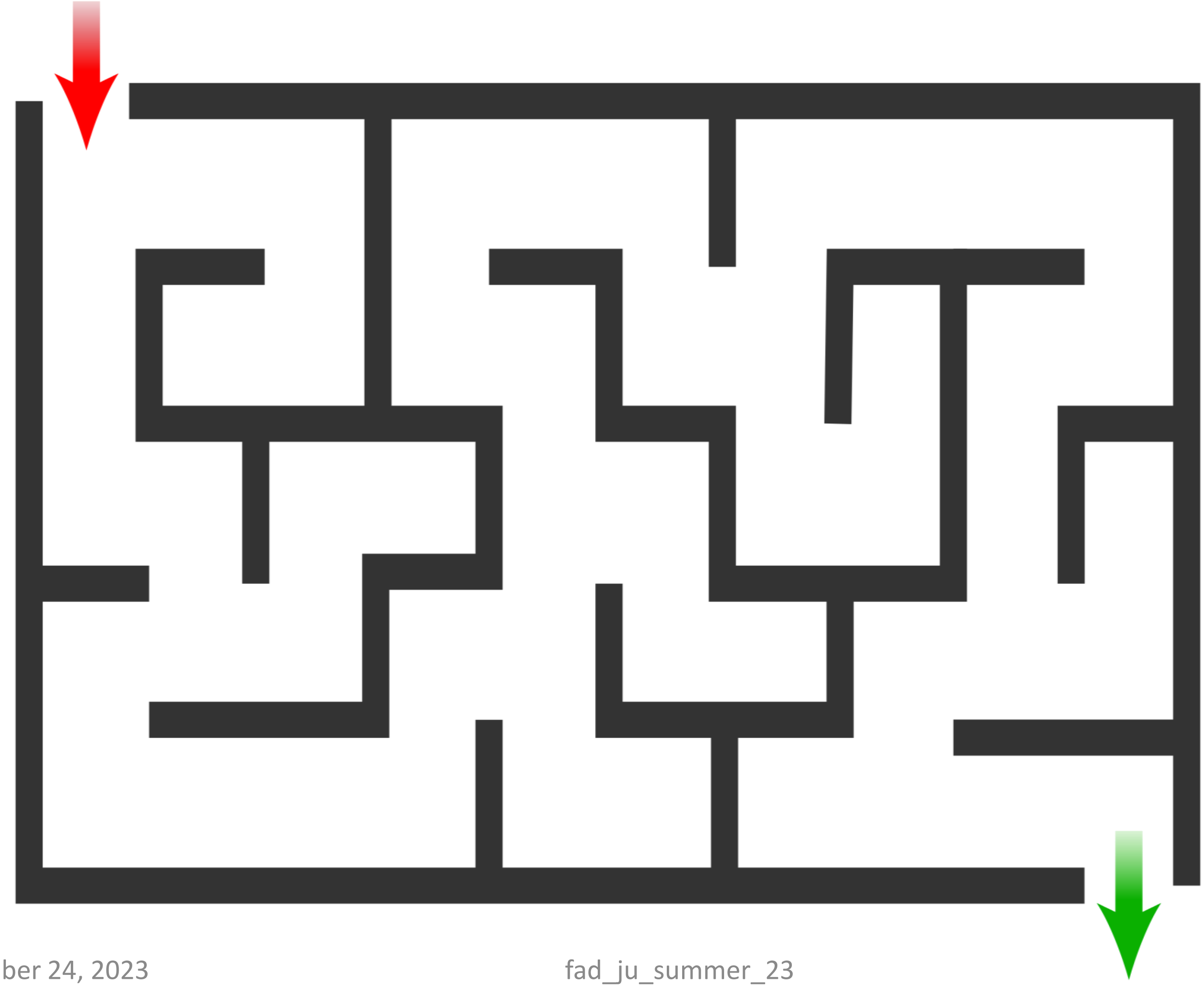




UNSUPERVISED LEARNING

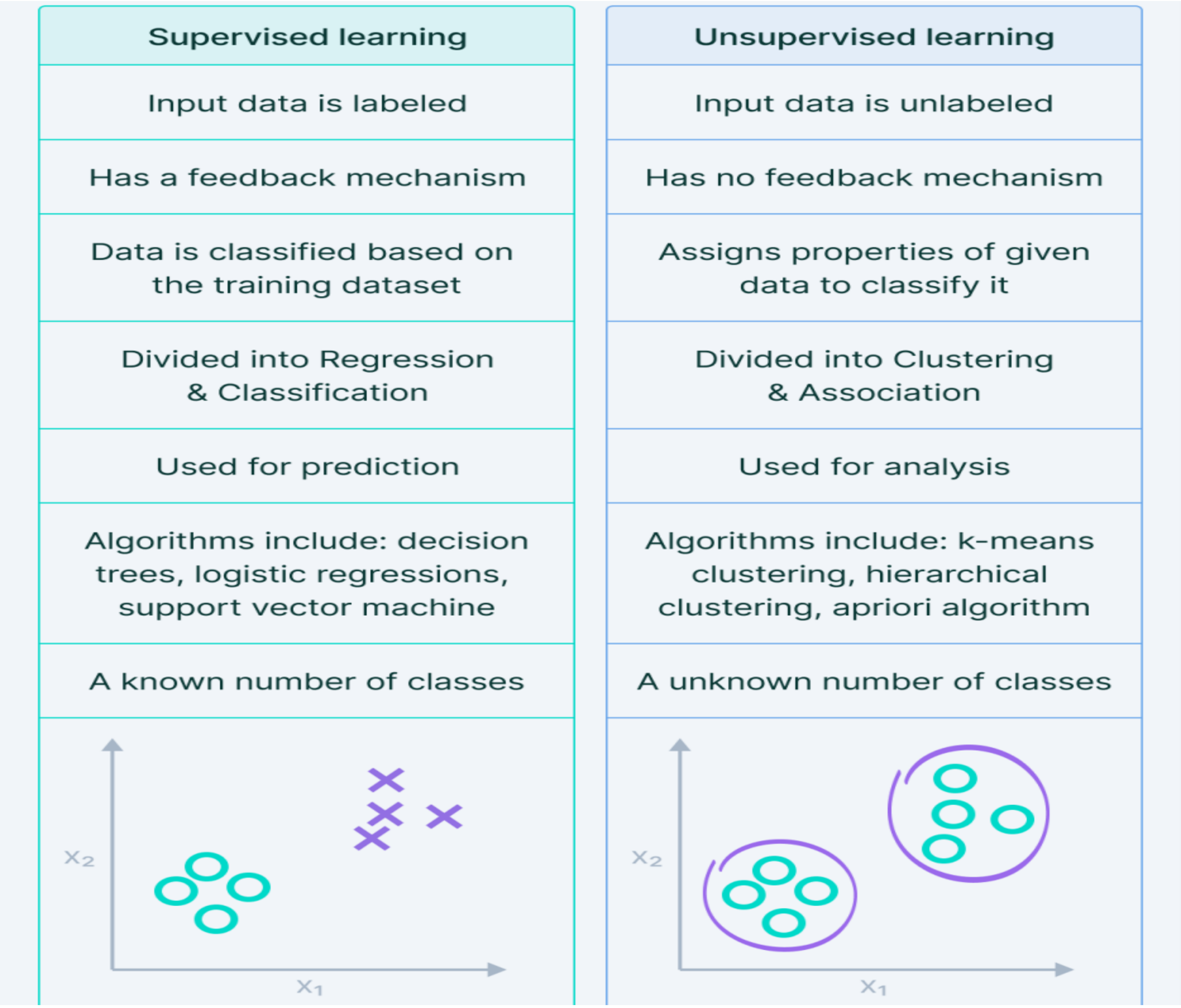
Unsupervised learning is a type of machine learning where the algorithm learns from unlabeled data without any predefined outputs or target variables.

Reinforcement Learning

- 

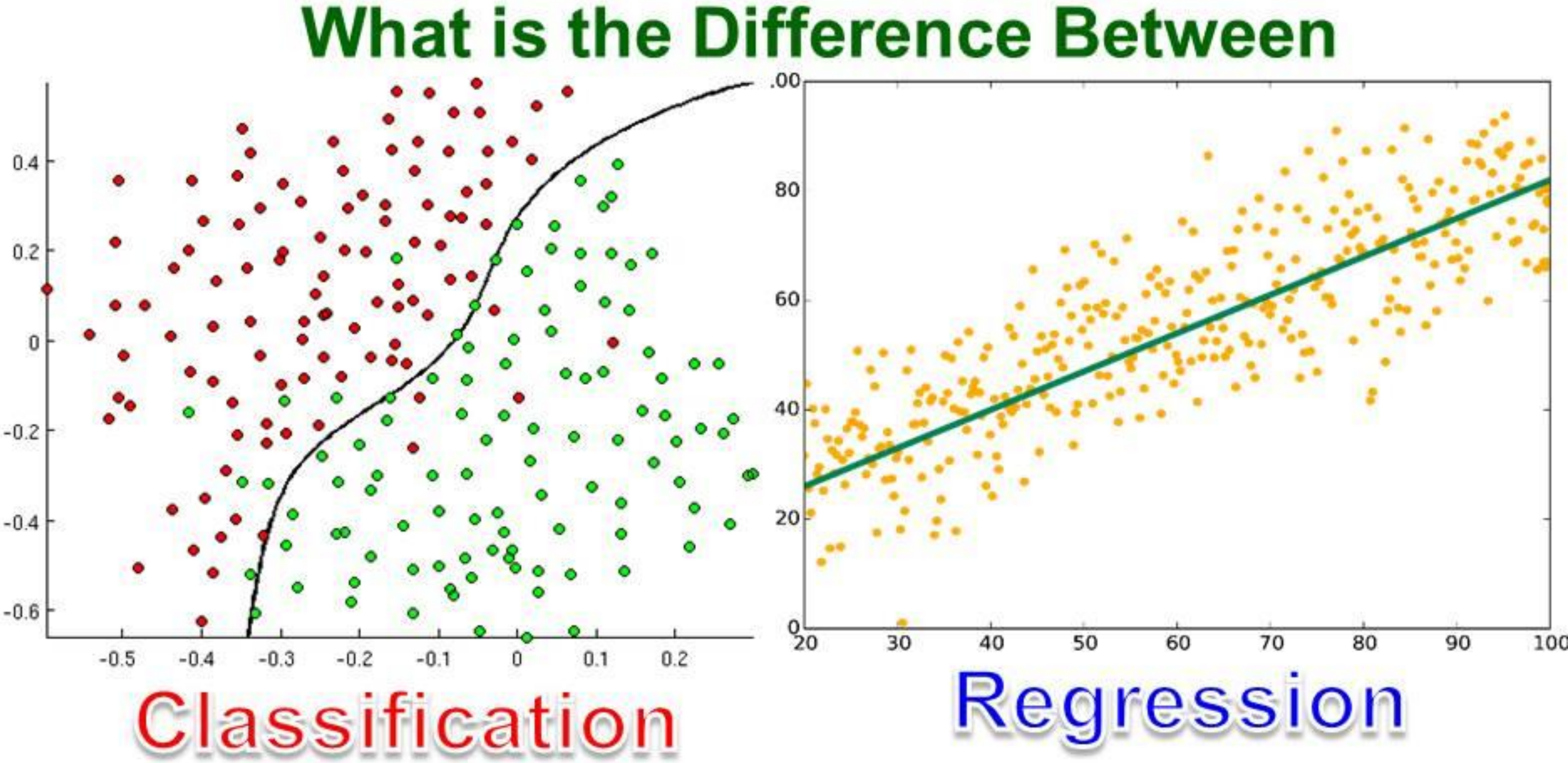
Supervised vs unsupervised learning

- .

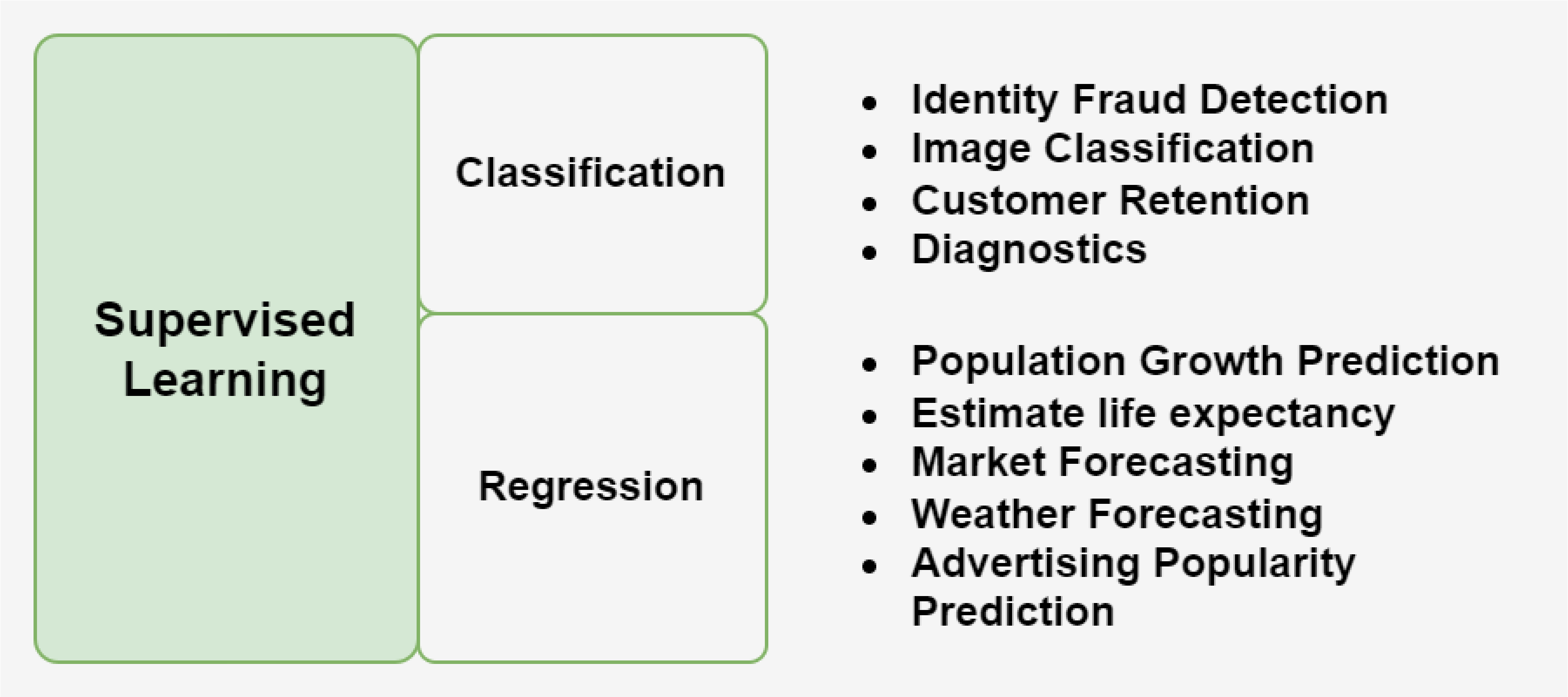


Supervised learning	Unsupervised learning
Input data is labeled	Input data is unlabeled
Has a feedback mechanism	Has no feedback mechanism
Data is classified based on the training dataset	Assigns properties of given data to classify it
Divided into Regression & Classification	Divided into Clustering & Association
Used for prediction	Used for analysis
Algorithms include: decision trees, logistic regressions, support vector machine	Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm
A known number of classes	A unknown number of classes

Classification vs Regression



Applications of Supervised Learning



Supervised Learning

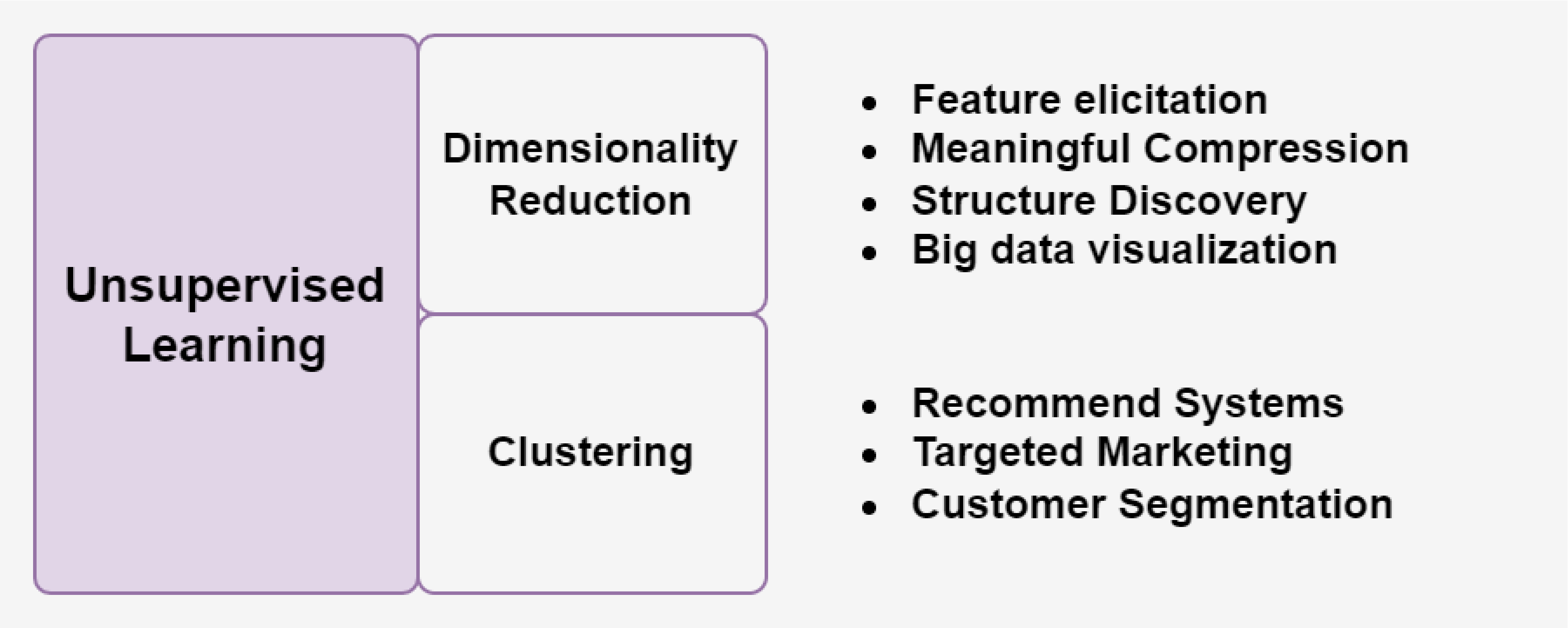
Classification

- Identity Fraud Detection
- Image Classification
- Customer Retention
- Diagnostics

Regression

- Population Growth Prediction
- Estimate life expectancy
- Market Forecasting
- Weather Forecasting
- Advertising Popularity Prediction

Applications of Unsupervised Learning



Unsupervised Learning

Dimensionality Reduction

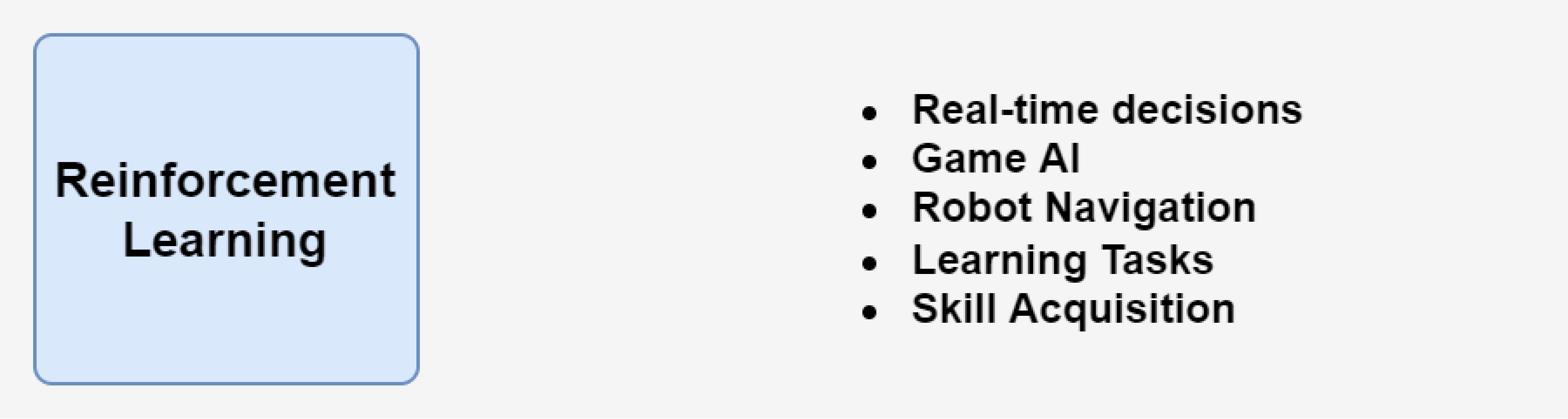
- Feature elicitation
- Meaningful Compression
- Structure Discovery
- Big data visualization

Clustering

- Recommend Systems
- Targeted Marketing
- Customer Segmentation

Applications of Reinforcement

- .



Reinforcement Learning

- Real-time decisions
- Game AI
- Robot Navigation
- Learning Tasks
- Skill Acquisition

Unsupervised Learning

- Deals with unlabeled data
- Find hidden, useful and interesting patterns, similarities, dissimilarities
- Requires minimal human interaction.
- Models can perform more complex tasks than Supervised Learning models, but they are also more unpredictable.
- It is a powerful tool for discovering hidden patterns in data and gaining a deeper understanding of complex datasets without the need for labeled examples.

Unsupervised Learning Tasks

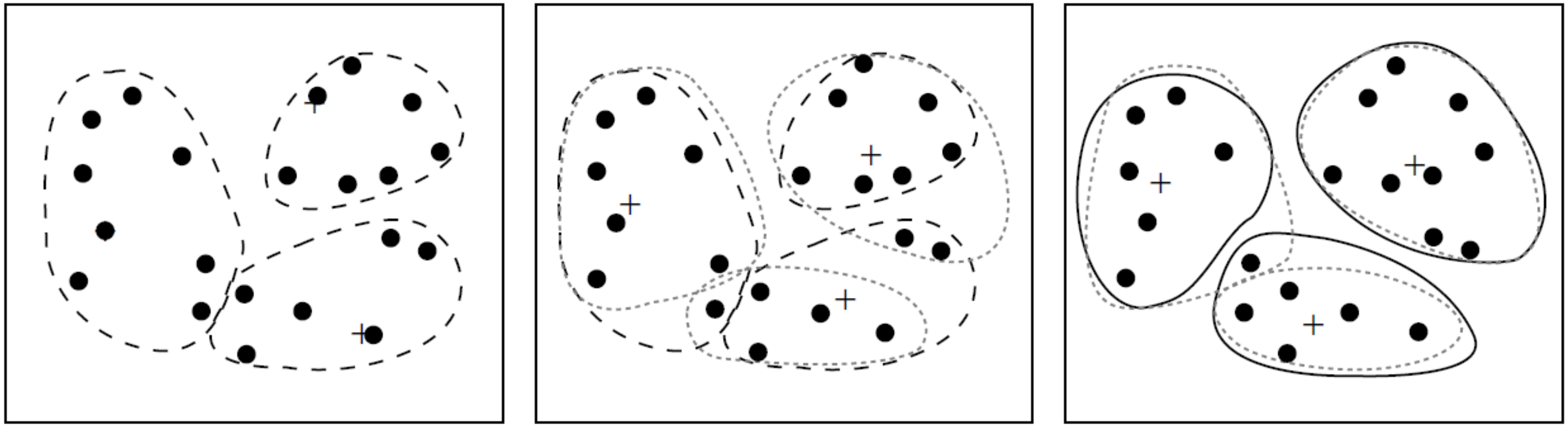
- The machine is trained on **unlabelled** data without any guidance.
- ***Clustering problem*** is to partition data into groups based on similarities
  - Example: targeted marketing from a list of customers and some information about them
- ***Association problems*** involve discovering patterns in data finding co-occurrences and so on.
  - Example: relationship between bread and jam
- ***Anomaly detection*** is used for tracking unusual activities
  - Example: Credit card fraud- unusual behaviour
- **Dimension Reduction**
  - Visualization of high-dimensional data can be achieved through dimensionality reduction

Why Unsupervised Learning?

- Unsupervised machine learning finds unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

Unsupervised Learning: Clustering

- Find hidden patterns in the data and groups unlabeled data based on their similarities or differences.
- These patterns can relate to the shape, size, or color and are used to group data items or create clusters.
- Clustering algorithms can be categorized into a few types, specifically exclusive, overlapping, hierarchical, and probabilistic.



(a) Initial clustering (b) Iterate (c) Final clustering

Clustering

- Clustering is a process of grouping similar objects together; i.e., to partition unlabeled examples into disjoint subsets of clusters, such that:
  - Examples within a cluster are similar (in this case, we speak of *high intraclass similarity*).
  - Examples in different clusters are different (in this case, we speak of *low interclass similarity*).
- Two kinds of inputs can be used for grouping:
  - (a) in *similarity-based clustering*, the input to the algorithm is an $n \times n$ *dissimilarity matrix* or *distance matrix*;
  - (b) in *feature-based clustering*, the input to the algorithm is an $n \times D$ *feature matrix* or *design matrix*, where n is the number of examples in the dataset and D the dimensionality of each sample.
- Similarity-based clustering allows easy inclusion of domain-specific similarity, while feature-based clustering has the advantage that it is applicable to potentially noisy data.

Clustering [cont'd]

- Several questions regarding the clustering process arise.
  - What is a natural grouping among the objects? We need to define the "groupness" and the "similarity/distance" between data.
  - How can we group samples? What are the best procedures?
  - How many clusters should we look for in the data? Shall we state this number a priori?
  - What constitutes a good grouping? What objective measures can be defined to evaluate the quality of the clusters?
- There is not always a single or optimal answer to these questions.
- It used to be said that **clustering is a "subjective" issue**.
- Clustering will help us to **describe, analyze, and gain insight** into the data, **but the quality of the partition depends to a great extent on the application and the analyst**.

Similarity and Distances

- To speak of similar and dissimilar data, we need to introduce a notion of the similarity of data. A simple way for modeling similarity is by means of a Gaussian kernel:

$$s(a, b) = e^{-\gamma d(a,b)}$$

  - where *d(a, b)* is a metric function, and γ is a constant that controls the decay of the function.
  - Observe that when *a* = *b*, the similarity is maximum and equal to one.
  - On the contrary, when *a* is very different to *b*, the similarity tends to zero.
- The former modeling of the similarity function suggests that we can use the notion of distance as a surrogate. The most widespread distance metric is the *Minkowski distance*:

$$d(a, b) = (\sum_{i=1}^{d} |a_i - b_i|^p)^{1/p}$$

  - where *d(a, b)* stands for the distance between two elements *a, b* ∈ R*d* ,
  - *d* is the dimensionality of the data, and *p* is a parameter.
- The best-known instantiations of this metric are as follows:
  - when *p* = 2, we have the *Euclidean distance*,
  - when *p* = 1, we have the *Manhattan distance*, and
  - when *p* = inf, we have the *max-distance*. In this case, the distance corresponds to the component |*ai* − *bi*| with the highest value.

Types of Clustering Techniques

1. *Partitional algorithms*: these start with a random partition and refine it iteratively. Also known as “flat” clustering.
   – For example, K-means clustering
2. *Hierarchical algorithms*: these organize the data into hierarchical structures, where data can be agglomerated in the bottom-up direction, or split in a top-down manner.
   – *For example, agglomerative clustering*

K-means Clustering

- K-means algorithm is a hard partition algorithm with the goal of assigning each data point to a single cluster.
- K-means algorithm divides a set of *n* samples *X* into *k* disjoint clusters *ci, i* = 1*, . . . , k*, each described by the mean *µi* of the samples in the cluster. The means are commonly called cluster *centroids*.
- The K-means algorithm assumes that all *k* groups have equal variance. K-means clustering solves the following minimization problem:

$$\arg min_c \sum_{j=1}^{k} \sum_{x \in c_j} d(x, \mu_j) = \arg min_c \sum_{j=1}^{k} \sum_{x \in c_j} ||x - \mu_j||_2^2$$

$$inertia = \sum_{i=0}^{n} min_{\mu_j \in c}(||x_i - \mu_j||^2))$$

- where *ci* is the set of points that belong to cluster *i* and *µi* is the center of the class *ci*.
- K-means clustering objective function uses the square of the Euclidean distance

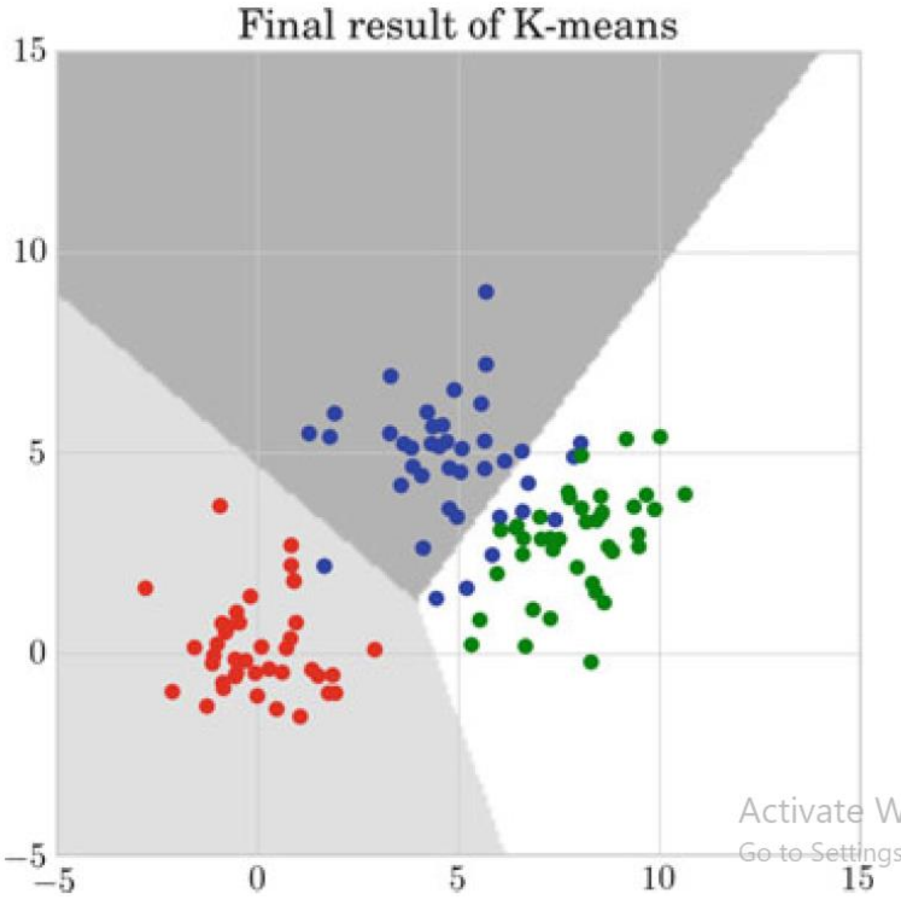
d(x, µj) = $||x - \mu_j||^2$, that is also referred to as the *inertia* or *within-cluster sumof- squares*.

K-means clustering algorithm

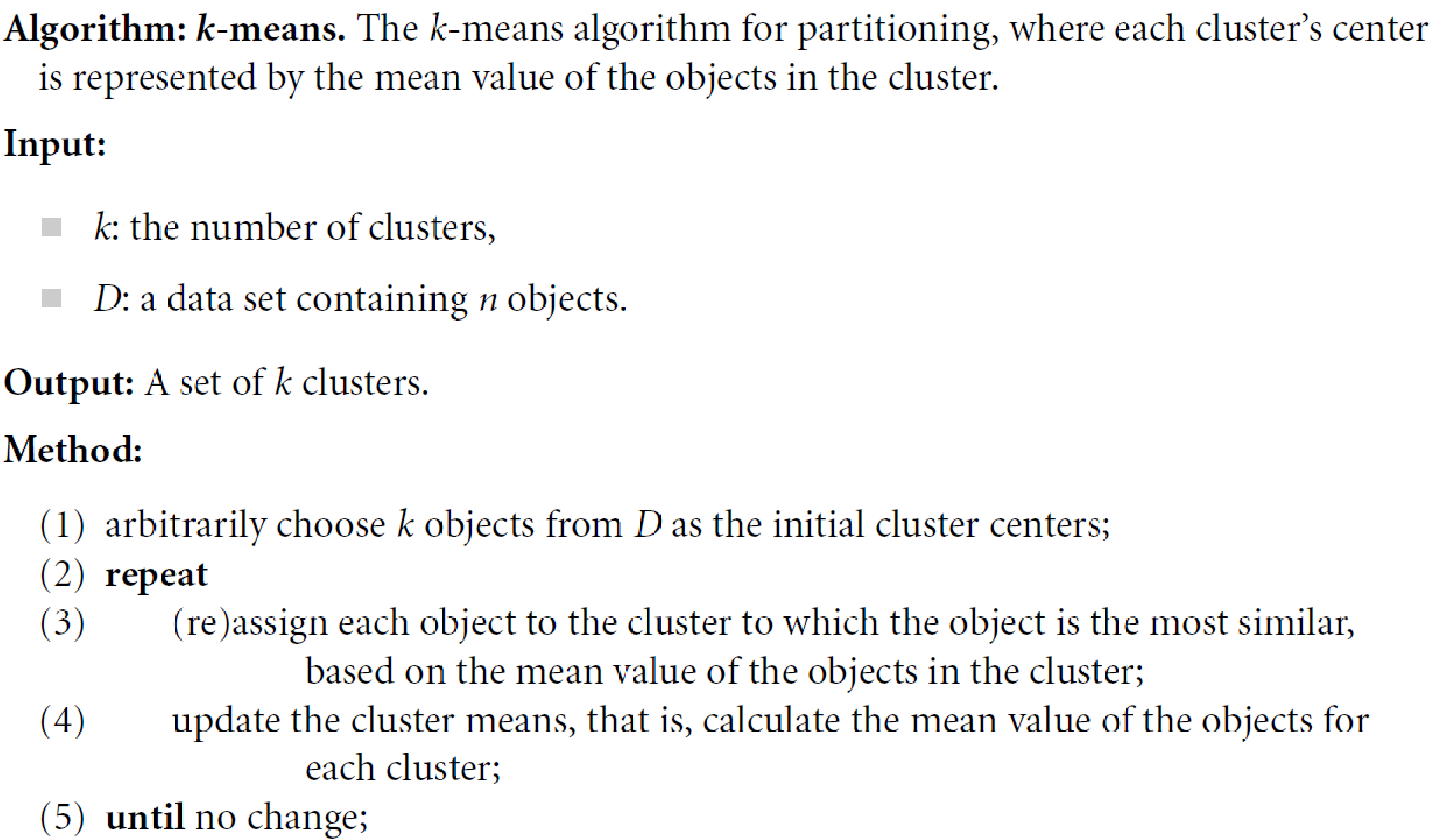
- *The K-means algorithm*, also known as Lloyd's algorithm, is an iterative procedure that searches for a solution of the *K*-means clustering problem and works as follows.
- First, we need to decide the number of clusters, *k*. Then we apply the following procedure:

1. Initialize (e.g., randomly) the *k* cluster centers, called *centroids*.
2. Decide the class memberships of the *n* data samples by assigning them to the nearest-cluster centroids (e.g., the center of gravity or mean).
3. Re-estimate the *k* cluster centers, *ci*, by assuming the memberships found above are correct.
4. If none of the *n* objects changes its membership from the last iteration, then exit. Otherwise go to step 2.

Fig. 7.2 Original samples (*dots*) generated by three distributions and the partition of the space according to the K-means clustering



K means clustering Algorithm



Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

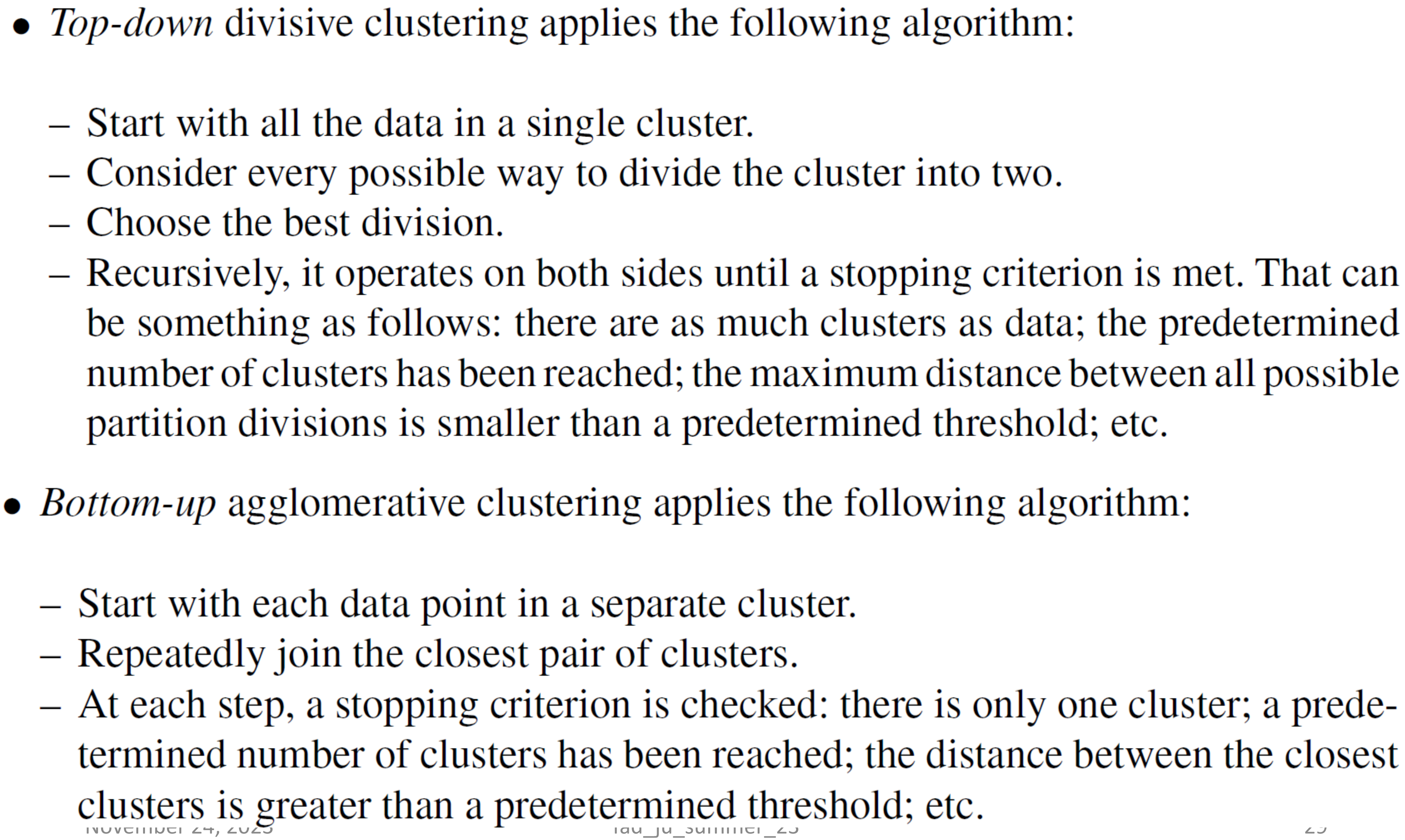
Method:

```
(1) arbitrarily choose $k$ objects from $D$ as the initial cluster centers;
(2) repeat
(3)     (re)assign each object to the cluster to which the object is the most similar,
            based on the mean value of the objects in the cluster;
(4)     update the cluster means, that is, calculate the mean value of the objects for
            each cluster;
(5) until no change;
```

Hierarchical Clustering

- In general, there are two types of hierarchical clustering:
  - *Top-down* divisive clustering algorithm:
  - *Bottom-up* agglomerative clustering algorithm:
- for the agglomerative clustering. The linkage criterion determines the metric used for the cluster merging strategy:
  - *Maximum* or *complete* linkage minimizes the maximum distance between observations of pairs of clusters. Based on the similarity of the two least similar members of the clusters, this clustering tends to give tight spherical clusters as a final result.
  - *Average* linkage averages similarity between members, i.e., minimizes the average of the distances between all observations of pairs of clusters.
  - *Ward* linkage minimizes the sum of squared differences within all clusters. It is thus a variance-minimizing approach and in this sense is similar to the K-means objective function, but tackled with an agglomerative hierarchical approach.

Top-down and Bottom-up clustering algorithm



- *Top-down* divisive clustering applies the following algorithm:

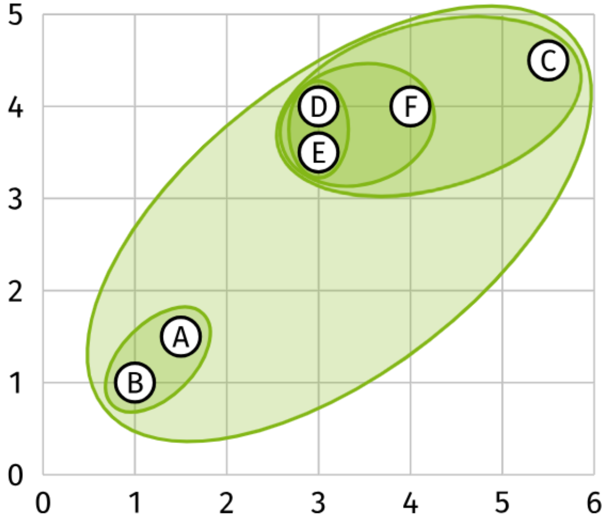
  – Start with all the data in a single cluster.
  – Consider every possible way to divide the cluster into two.
  – Choose the best division.
  – Recursively, it operates on both sides until a stopping criterion is met. That can be something as follows: there are as much clusters as data; the predetermined number of clusters has been reached; the maximum distance between all possible partition divisions is smaller than a predetermined threshold; etc.

- *Bottom-up* agglomerative clustering applies the following algorithm:

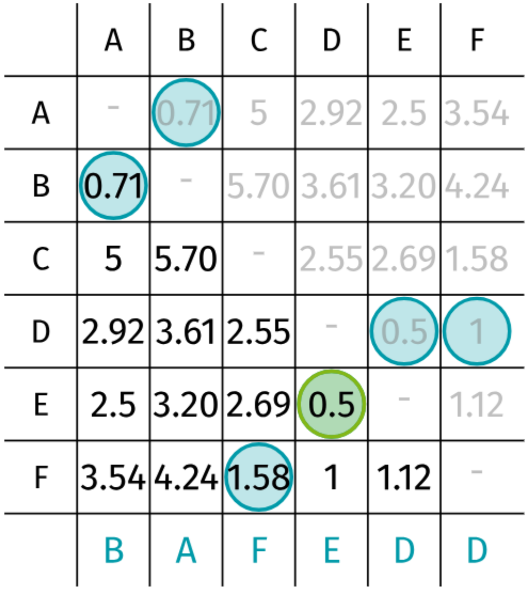
  – Start with each data point in a separate cluster.
  – Repeatedly join the closest pair of clusters.
  – At each step, a stopping criterion is checked: there is only one cluster; a predetermined number of clusters has been reached; the distance between the closest clusters is greater than a predetermined threshold; etc.

Example: Hierarchical Clustering

-

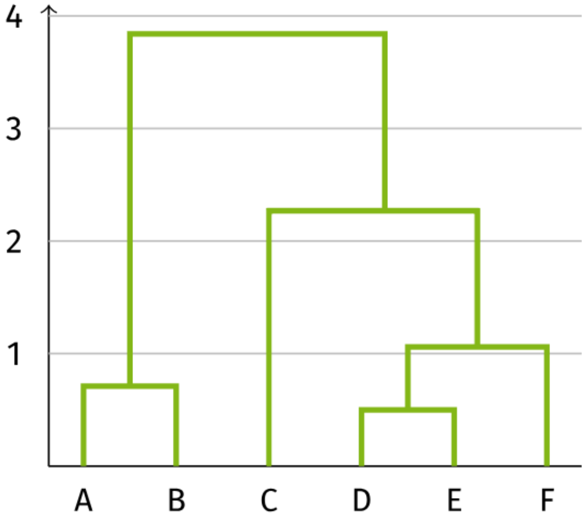


scatter plot



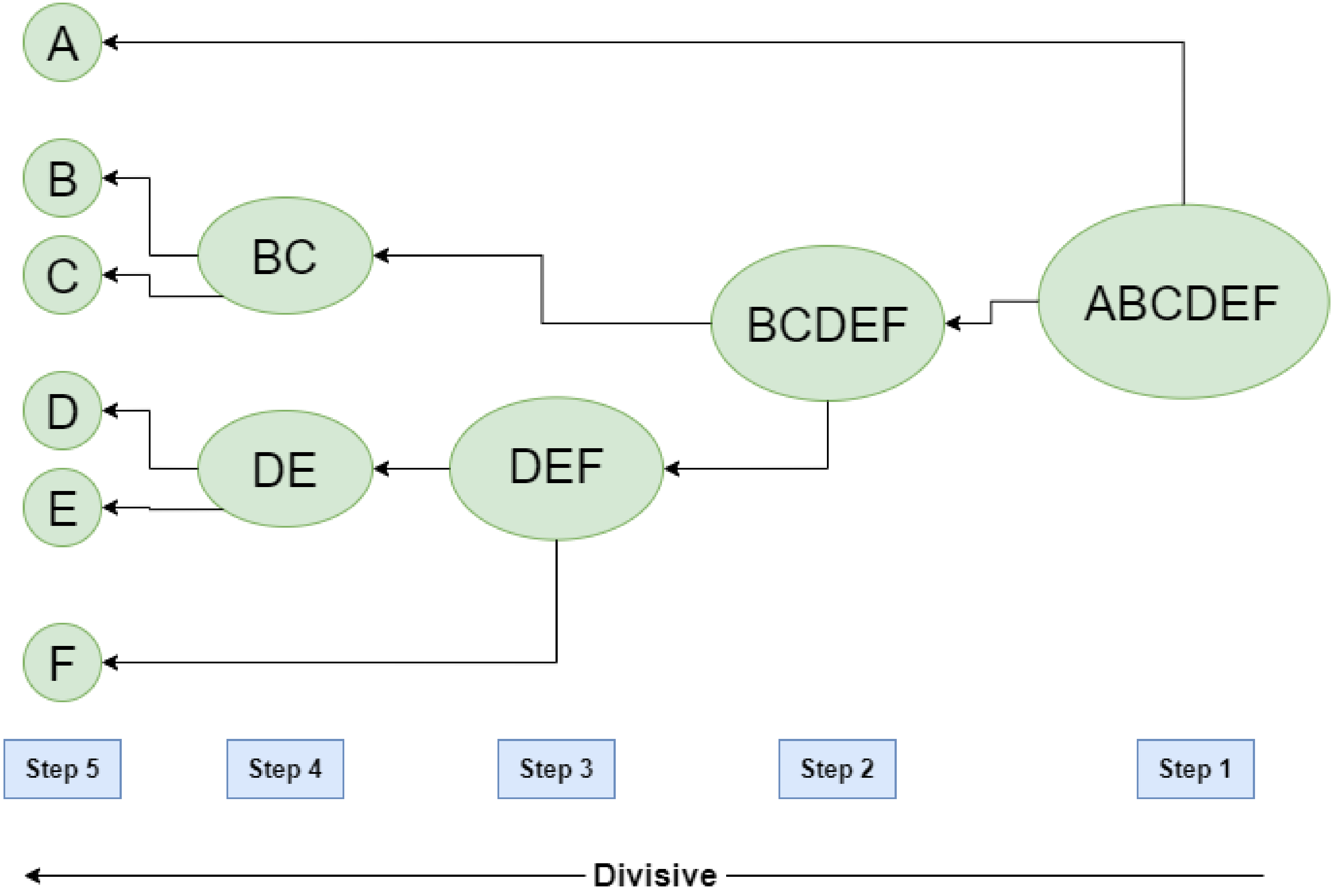
	A	B	C	D	E	F
A	-	0.71	5	2.92	2.5	3.54
B	0.71	-	5.70	3.61	3.20	4.24
C	5	5.70	-	2.55	2.69	1.58
D	2.92	3.61	2.55	-	0.5	1
E	2.5	3.20	2.69	0.5	-	1.12
F	3.54	4.24	1.58	1	1.12	-
	B	A	F	E	D	D

distance matrix

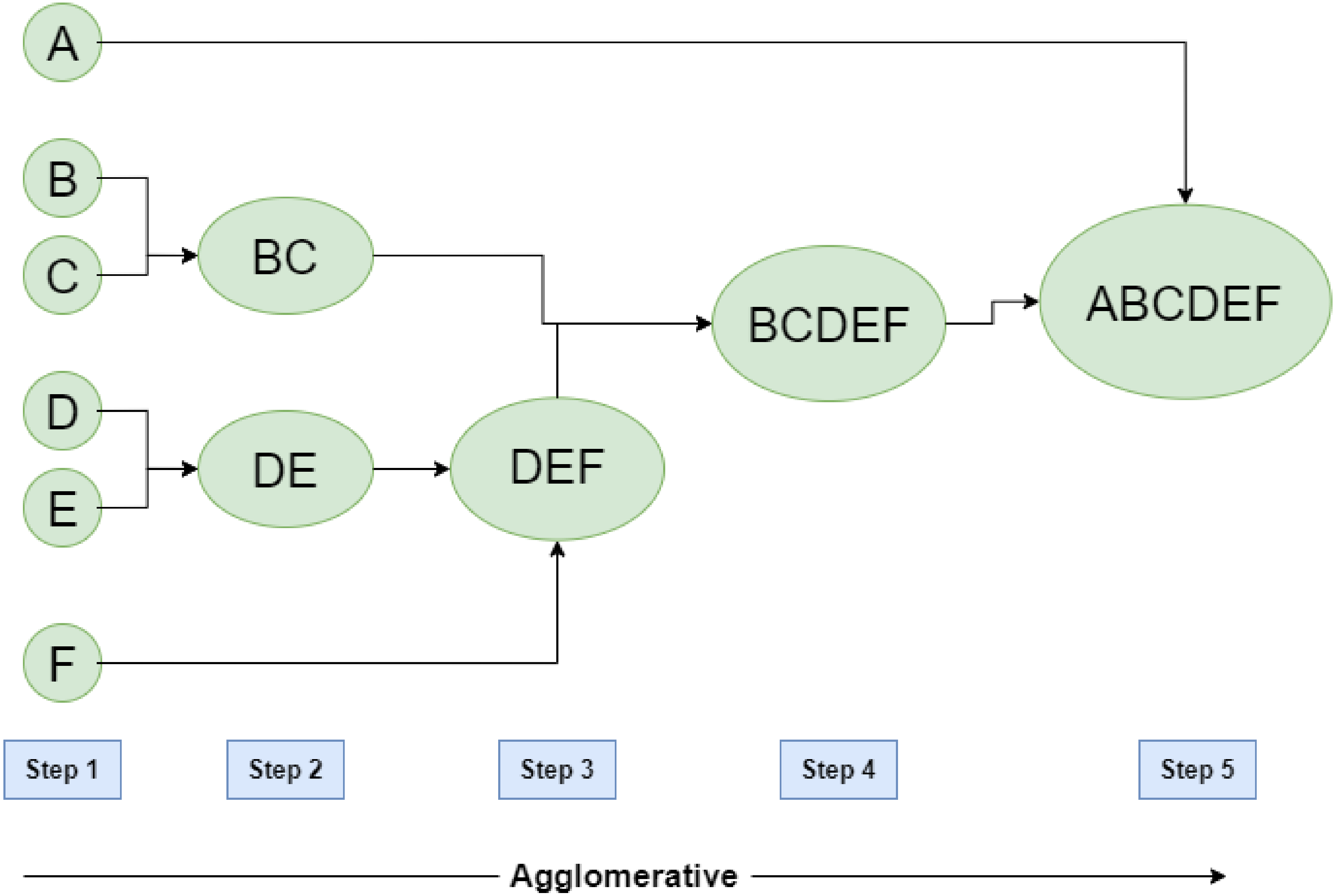


dendrogram

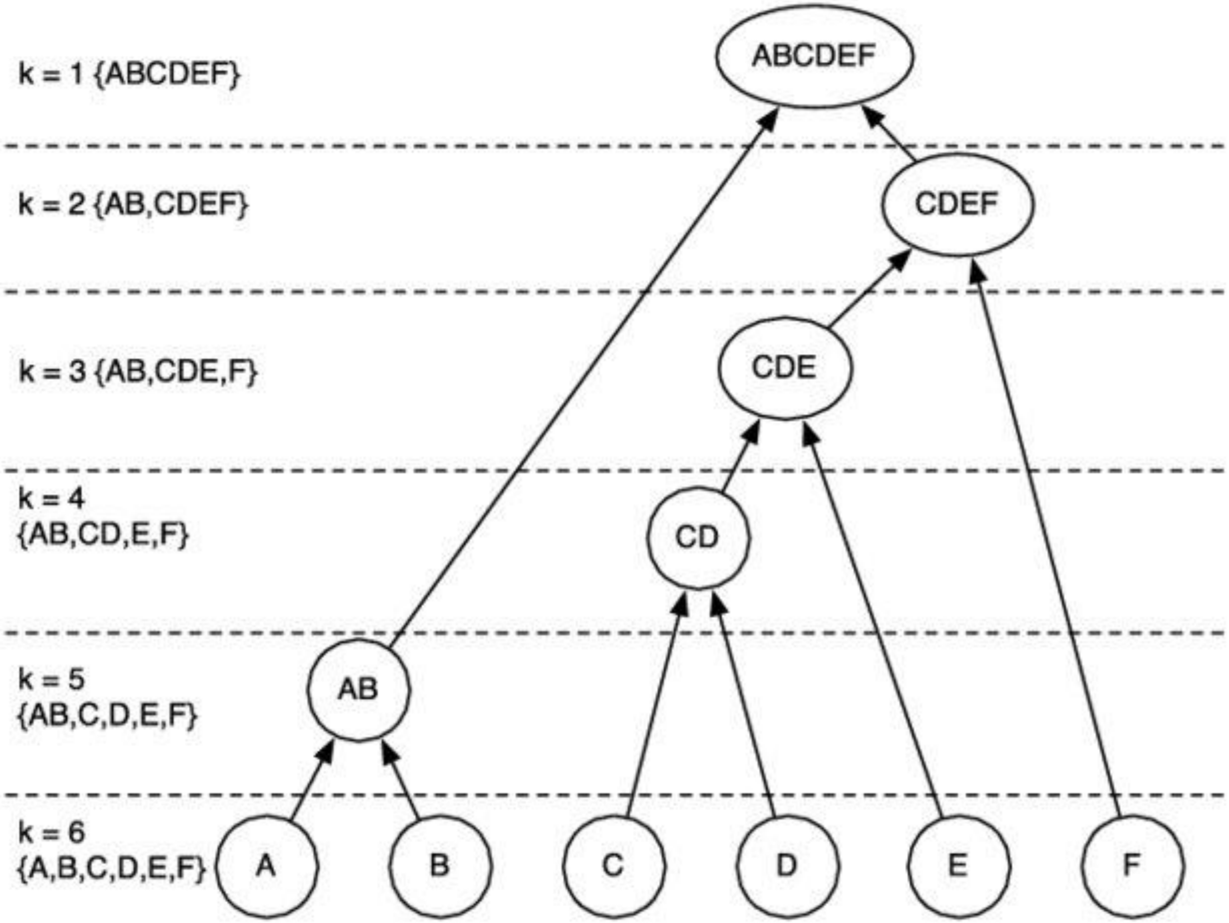
Top-down Clustering



Bottom-up clustering

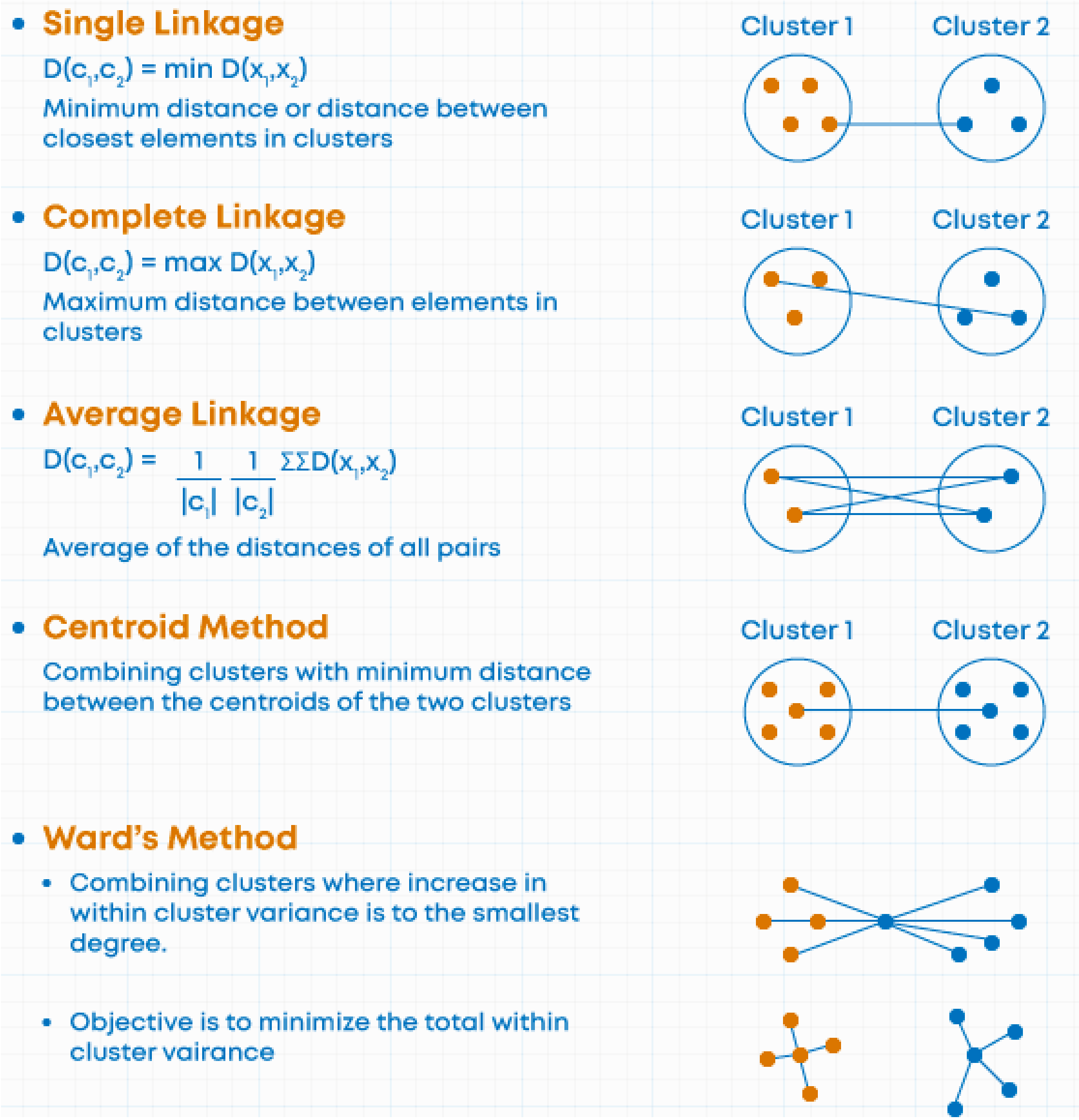


- ,


k = 1 {ABCDEF}
k = 2 {AB,CDEF}
k = 3 {AB,CDE,F}
k = 4
{AB,CD,E,F}
k = 5
{AB,C,D,E,F}
k = 6
{A,B,C,D,E,F}
ABCDEF
CDEF
CDE
CD
AB
A
B
C
D
E
F

Distances for Hierarchical Clustering

- .



- **Single Linkage**

  $D(c_1,c_2) = \min D(x_1,x_2)$

  Minimum distance or distance between closest elements in clusters

- **Complete Linkage**

  $D(c_1,c_2) = \max D(x_1,x_2)$

  Maximum distance between elements in clusters

- **Average Linkage**

  $$D(c_1,c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\Sigma\Sigma D(x_1,x_2)$$

  Average of the distances of all pairs

- **Centroid Method**

  Combining clusters with minimum distance between the centroids of the two clusters

- **Ward's Method**
  - Combining clusters where increase in within cluster variance is to the smallest degree.
  - Objective is to minimize the total within cluster vairance

Metrics to Measure Clustering Quality

- Rand Index
  - comparing the results in different clustering techniques
  - its value is between 0 and 1, larger values are desirable.
- Completeness
  - **all members of a given class are assigned to the same** cluster.
  - scores have real positive values between 0.0 and 1.0, larger values being desirable
- Homogeneity
  - each cluster contains only members of **a single class.**
  - scores have real positive values between 0.0 and 1.0, larger values being desirable
- V-measure
  - takes into account the completeness as well as the homogeneity
- Silhouette Coefficient
  - evaluates the compactness of the results of applying a specific clustering approach.

Rand Index

- The Rand index evaluates the similarity between two results of data clustering.
- Since in unsupervised clustering, class labels are not known, we use the Rand index to compare the coincidence of different clusterings obtained by different approaches or criteria.

Rand Index

- Given a set of n elements $S = \{o_1, . . . , o_n\}$, we can compare two partitions of $S1$: $X = \{X1, . . . , Xr\}$, a partition of S into r subsets; and $Y = \{Y1, . . . , , Ys\}$, a partition of S into s subsets. Let us use the annotations as follows:
  - a is the number of pairs of elements in S that are in the same subset in both X and Y ;
  - b is the number of pairs of elements in S that are in different subsets in both X and Y ;
  - c is the number of pairs of elements in S that are in the same subset in X , but in different subsets in Y ; and
  - d is the number of pairs of elements in S that are in different subsets in X , but in the same subset in Y .
- The Rand index, RI, is defined as follows:

$$RI = (a + b)/ (a + b + c + d)$$

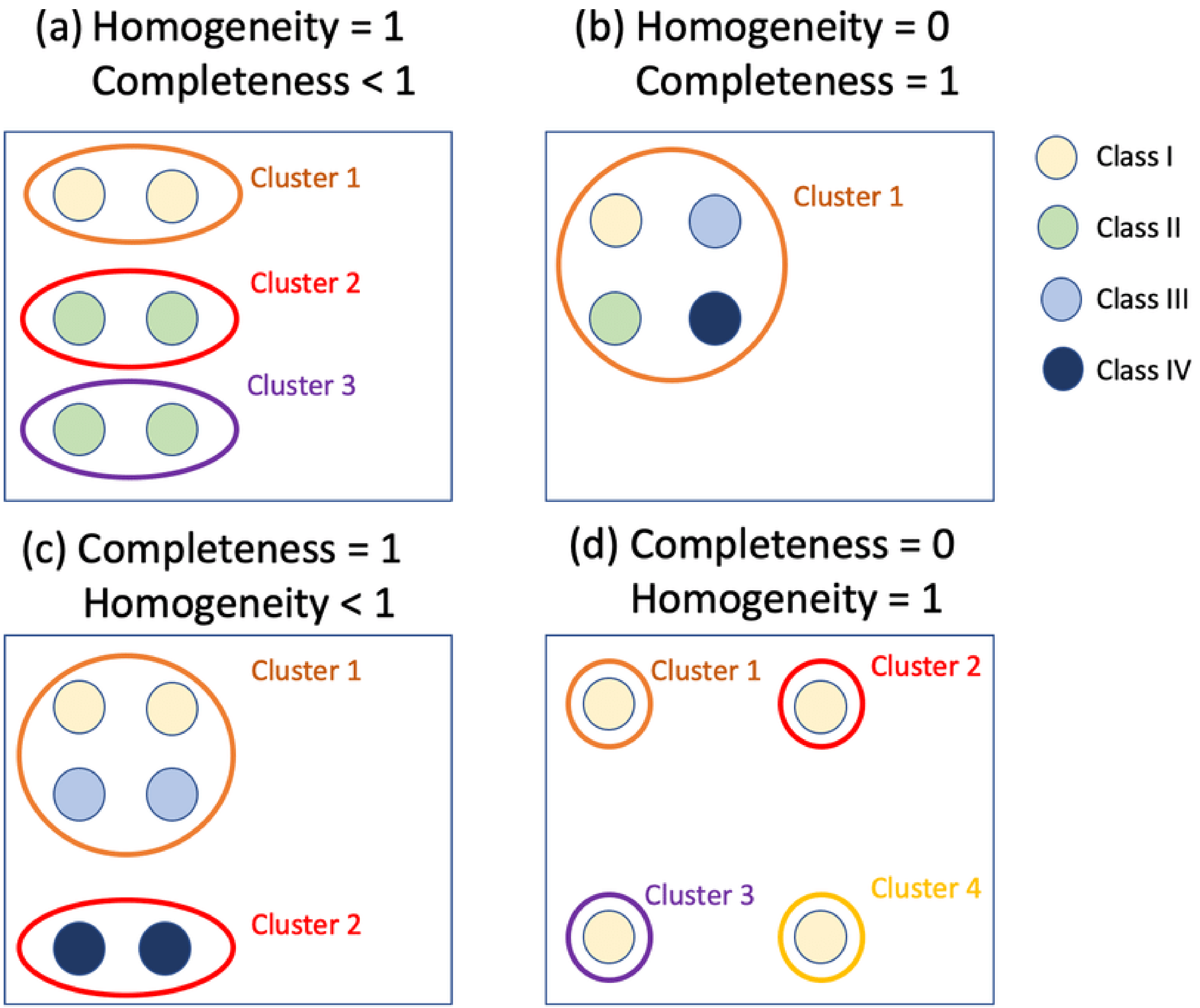
- its value is between 0 and 1.

Homogeneity and Completeness

- We say that a clustering result satisfies a *homogeneity* criterion if all of its clusters contain only data points which are members of the same original (single) class.
- A clustering result satisfies a *completeness* criterion if all the data points that are members of a given class are elements of the same predicted cluster.
- Note that both scores have real positive values between 0.0 and 1.0, larger values being desirable.

Homogeneity and Completeness

- .



V-measure Scores

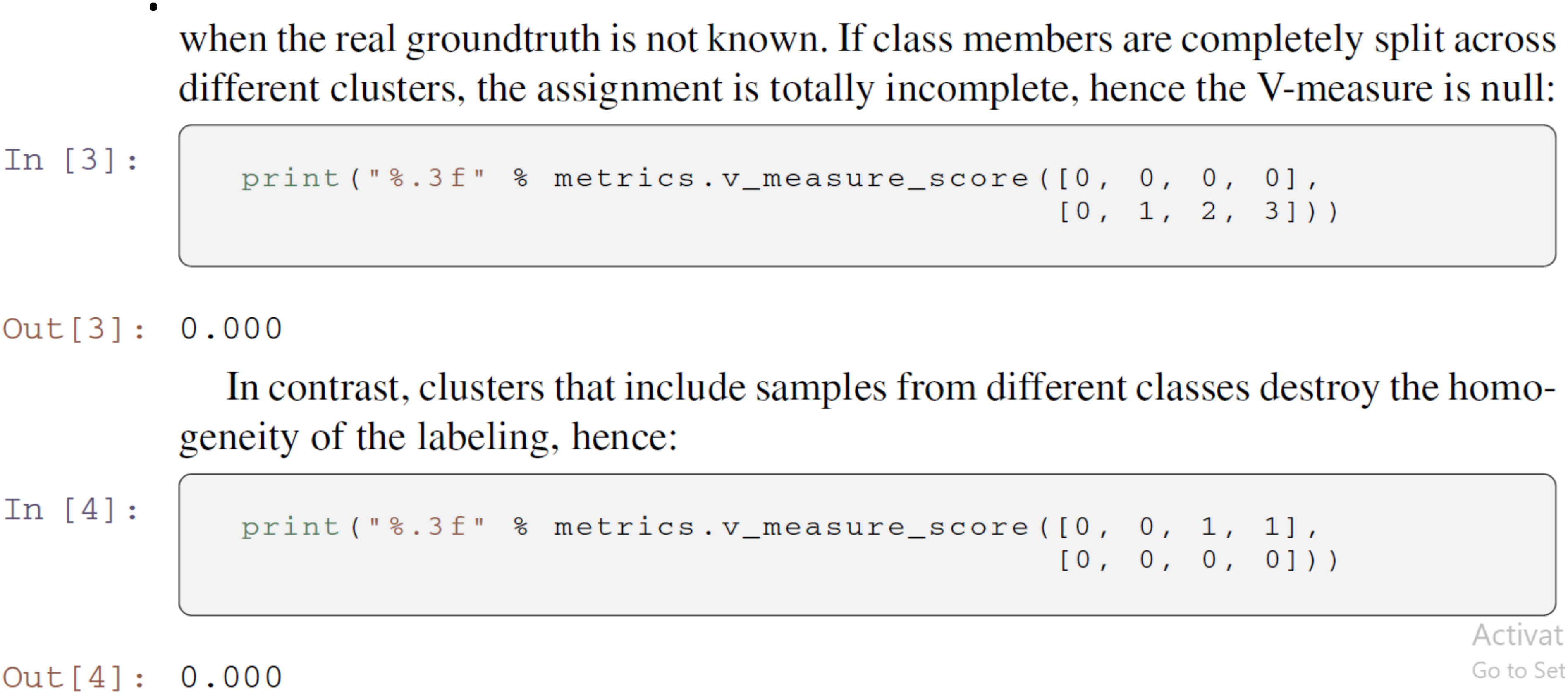
- The *V-measure* is the harmonic mean between the homogeneity and the completeness defined as follows:

v=2∗(homogeneity∗completeness)/(homogeneity+completeness)

- The V-measure has bounded scores:
  - 0.0 means the clustering is extremely bad;
  - 1.0 indicates a perfect clustering result.
- It can be interpreted easily:
  - when analyzing the V-measure, low completeness or homogeneity explain in which direction the clustering is not performing well.
- Furthermore, we do not assume anything about the cluster structure.

- Note that this metric is not dependent of the absolute values of the labels: a permutation of the class or cluster label values will not change the score value in any way.
- Moreover, the metric is symmetric with respect to switching between the predicted and the original cluster label.
- This is very useful to measure the agreement of two independent label assignment strategies applied to the same dataset even when the real ground truth is not known.
- If class members are completely split across different clusters, the assignment is totally incomplete, hence the V-measure is null.
- In contrast, clusters that include samples from different classes destroy the homogeneity of the labeling, hence the V-measure is null.

- .



when the real groundtruth is not known. If class members are completely split across different clusters, the assignment is totally incomplete, hence the V-measure is null:

```
In [3]: print("%.3f" % metrics.v_measure_score([0, 0, 0, 0],
                                               [0, 1, 2, 3]))

Out[3]: 0.000
```

In contrast, clusters that include samples from different classes destroy the homogeneity of the labeling, hence:

```
In [4]: print("%.3f" % metrics.v_measure_score([0, 0, 1, 1],
                                               [0, 0, 0, 0]))

Out[4]: 0.000
```

Limitations of V-measure

- As a drawback, the metrics, homogeneity, completeness and hence, the V-measure are not normalized with regard to random labeling.
- This means that depending on the number of samples, clusters and groundtruth classes, a completely random labeling will not always yield the same values for homogeneity, completeness and hence, the V-measure.
- In particular, random labeling will not yield a zero score, and they will tend further from zero as the number of clusters increases.
- It can be shown that this problem can reliably be overcome when the number of samples is high, i.e., more than a thousand, and the number of clusters is less than 10.
- These metrics require knowledge of the groundtruth classes, while in practice this information is almost never available or requires manual assignment by human annotators.
- Instead, as mentioned before, these metrics can be used to compare the results of different clusterings.

Silhouette Score

- The Silhouette coefficient for a sample *i* can be written as follows:

  $$Silhouette(i) = (b - a)/ \max(a, b)$$

  - *a* is the intracluster distance of a sample in the dataset and
  - *b* the nearest cluster distance, for each sample.
- Silhouette coefficient is bounded **between −1 and +1**.
  - if the Silhouette *s(i)* is close to 0, it means that the sample is on the border of its cluster and the closest one from the rest of the dataset clusters.
  - A negative value means that the sample is closer to the neighbor cluster.
  - A high positive value, i.e., close to 1 would mean a compact cluster, and vice versa.
- The average of the Silhouette coefficients of all samples of a given cluster defines the "goodness" of the cluster.
- And the average of the Silhouette coefficients of all clusters gives idea of the quality of the clustering result.
- The Silhouette coefficient is generally **higher when clusters are compact**, when clusters are dense and well separated.

Rand Index vs Silhouette Coefficient

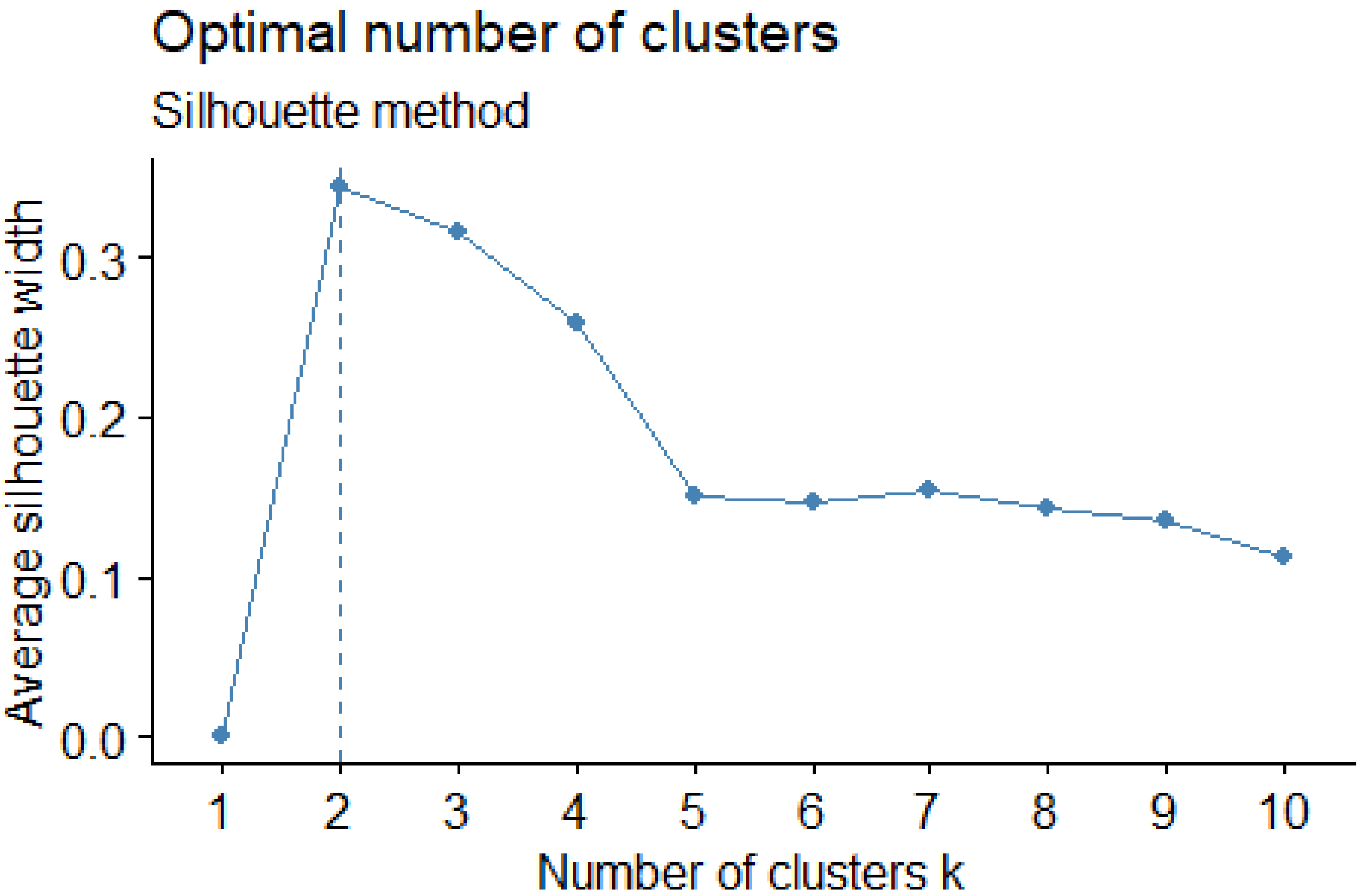
- Rand index or Rand measure compares the results in clustering techniques.
- *Silhouette coefficient* evaluates the compactness of the results of applying a specific clustering approach.

Finding Optimum Number of Cluster: K

- Silhouette Method
- Elbow Method

Optimal Number of Cluster: Silhouette method

- ‘

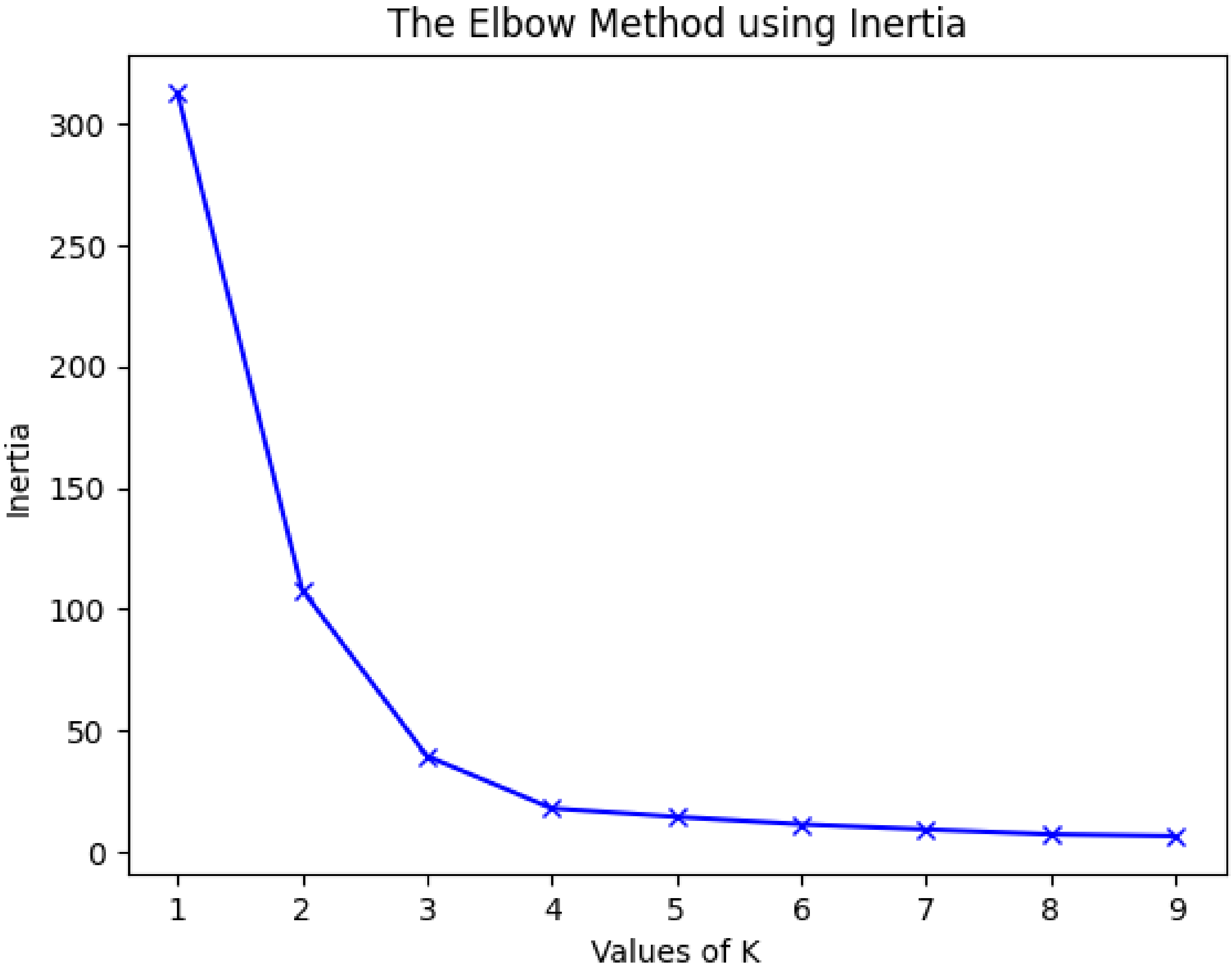


Elbow Method

- Elbow Method is a technique that we use to determine the number of centroids(k) to use in a k-means clustering algorithm. In this method to determine the k-value we continuously iterate for *k*=1 to *k=n* (Here *n* is the hyperparameter that we choose as per our requirement)
- For every value of k, we calculate the within-cluster sum of squares (WCSS) value.
- WCSS - It is defined as the sum of square distances between the centroids and each points.Now For determining the best number of clusters(k) we plot a graph of k versus their WCSS value.
- Surprisingly the graph looks like an elbow (*which we will see later*).
  - When *k*=1 the WCSS has the highest value but with increasing k value WCSS value starts to decrease.
  - We choose that value of k from where the graph starts to look like a straight line.

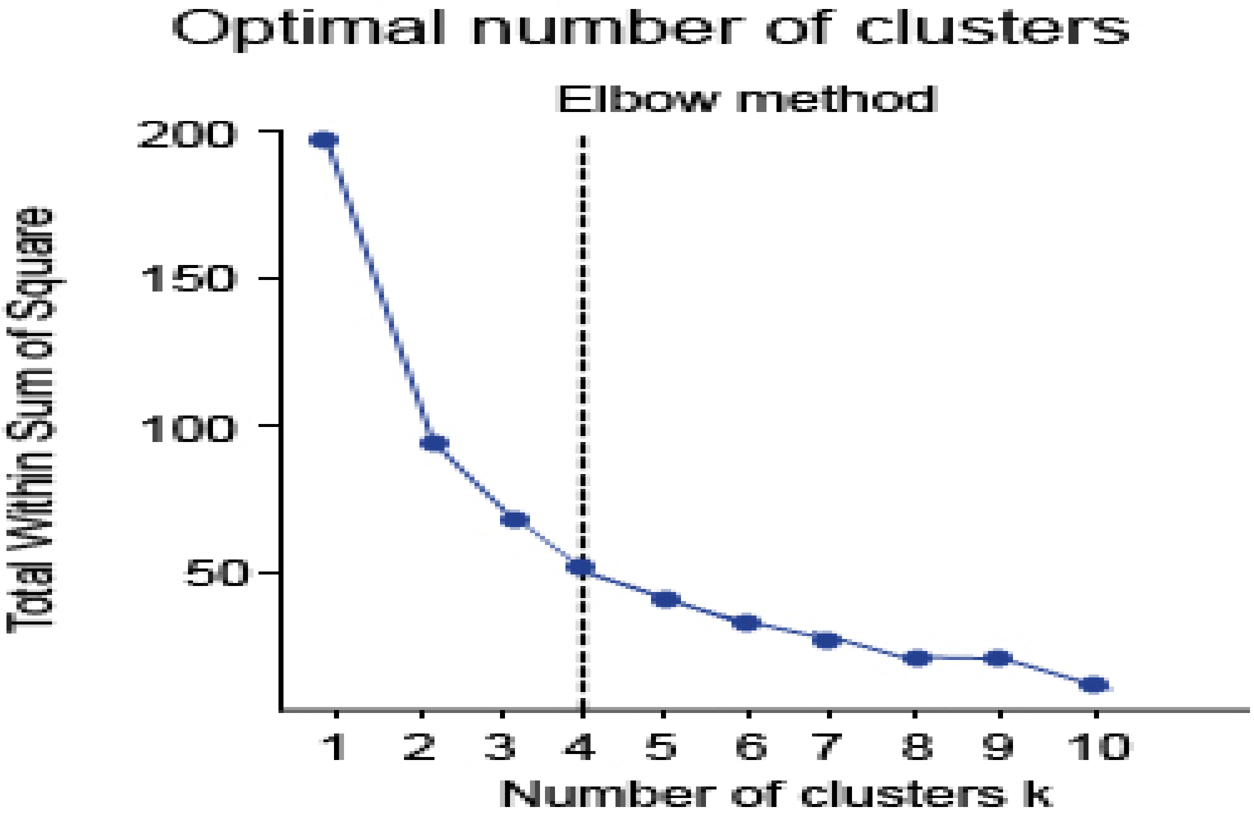
Optimal Number of Cluster: Elbow method

- .



Example: Elbow method

- .



Unsupervised learning Challenges

- Unsupervised learning presents challenges such as
  - determining the optimal number of clusters,
  - handling high-dimensional data, and
  - interpreting the discovered patterns.

Exploratory Data Analysis (EDA)

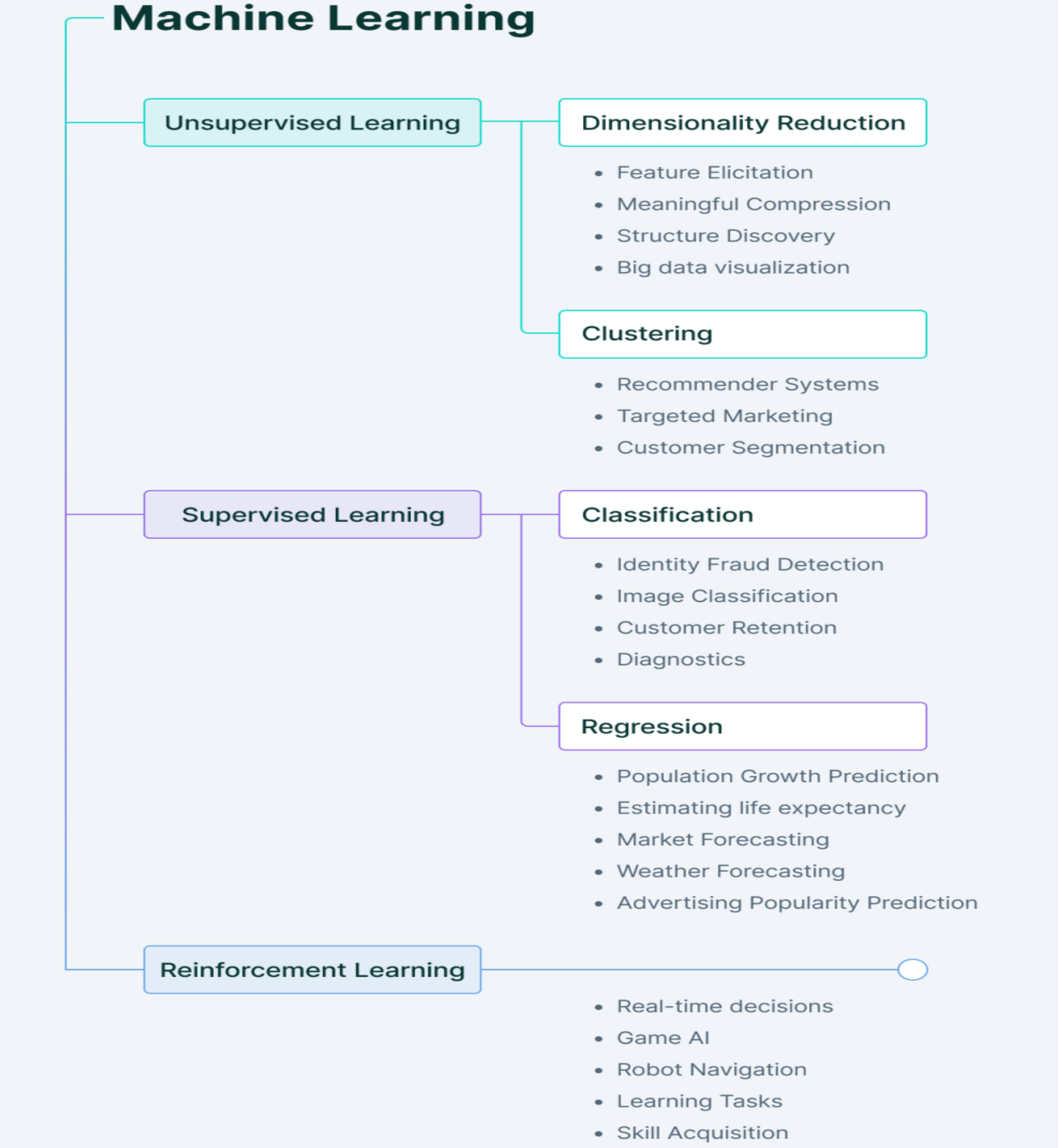
- Unsupervised learning is often used for exploratory data analysis, allowing researchers and data scientists to gain insights into the underlying structure of a dataset.

Pros & Cons of Unsupervised Learning

- Pros
  - Unsupervised Machine Learning is used for **complex tasks** with compare to Supervised Machine Learning. As it does not have labelled dataset like Supervised Machine Learning, it is used for many unlabelled datasets.
  - It is preferable as it is easy to get unlabelled datasets with compare to labelled datasets.
- Cons
  - UML is intrinsically more difficult with compare to supervised machine learning as it does not keep predefined output value.
  - Usually, the magnitude of accuracy is not high as supervised machine learning’s cases. Because it does not know about all of the input labels and exact output in advance.

Applications of Unsupervised Learning

- Anomaly Detection:
  - Anomaly detection uses unsupervised machine learning to find out unusual data-points in datasets. It is mainly useful in fraudulent detection.
- Recommender Systems:
  - Recommender system is one of the frequently used applications of machine learning. It is used in many internet ventures such as Netflix, Amazon Prime Video, Hotstar, Amazon, Flipkart, Alibaba, Youtube, Google Advertisements and other recommendation systems of e-commerce, entertainment websites or platforms.
- Customer Segmentation:
  - Customer segmentation is also one the great applications of machine learning for each kind of companies. Through a customer segmentation, businesses can find out their targeted-customers, can show them their products or services.
- Genetics:
  - DNA pattern recognition has become a usual stuff in lots of countries to find out the true parents. It is also used in research & development of many medical entities.



Machine Learning

Unsupervised Learning

Dimensionality Reduction

- Feature Elicitation
- Meaningful Compression
- Structure Discovery
- Big data visualization

Clustering

- Recommender Systems
- Targeted Marketing
- Customer Segmentation

Supervised Learning

Classification

- Identity Fraud Detection
- Image Classification
- Customer Retention
- Diagnostics

Regression

- Population Growth Prediction
- Estimating life expectancy
- Market Forecasting
- Weather Forecasting
- Advertising Popularity Prediction

Reinforcement Learning

- Real-time decisions
- Game AI
- Robot Navigation
- Learning Tasks
- Skill Acquisition

THANK YOU