1. What is the difference between supervised and unsupervised learning? Give some examples to illustrate your point.

Navigating the realm of machine learning, many grapple with understanding the key disparities between supervised and unsupervised learning. This article aims to elucidate these differences, addressing questions on input data, computational complexities, real-time analysis, and the reliability of results.

**Supervised learning**

When an algorithm is trained on a labelled dataset—that is, when the input data used for training is paired with corresponding output labels—it is referred to as [supervised learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/). Supervised learning aims to find a mapping or relationship between the input variables and the desired output, which enables the algorithm to produce precise predictions or classifications when faced with fresh, unobserved data.

An input-output pair training set is given to the algorithm during a supervised learning process. For every example in the training set, the algorithm iteratively modifies its parameters to minimize the discrepancy between its predicted output and the actual output (the ground truth). This procedure keeps going until the algorithm performs at an acceptable level.

Supervised learning can be divided into two main types:

1. **Regression:** In [regression](https://www.geeksforgeeks.org/types-of-regression-techniques/) problems, the goal is to predict a continuous output or value. For example, predicting the price of a house based on its features, such as the number of bedrooms, square footage, and location.
2. **Classification:** In [classification](https://www.geeksforgeeks.org/basic-concept-classification-data-mining/) problems, the goal is to assign input data to one of several predefined categories or classes. Examples include spam email detection, image classification (e.g., identifying whether an image contains a cat or a dog), and sentiment analysis.

**Why supervised learning?**

The basic aim is to approximate the mapping function(mentioned above) so well that when there is a new input data (x) then the corresponding output variable can be predicted. It is called supervised learning because the process of learning(from the training dataset) can be thought of as a teacher who is supervising the entire learning process. Thus, the “learning algorithm” iteratively makes predictions on the training data and is corrected by the “teacher”, and the learning stops when the algorithm achieves an acceptable level of performance (or the desired accuracy).

**Supervised Learning Example**

Suppose there is a basket which is filled with some fresh fruits, the task is to arrange the same type of fruits in one place. Also, suppose that the fruits are apple, banana, cherry, and grape. Suppose one already knows from their *previous work* (or experience) that, the shape of every fruit present in the basket so, it is easy for them to arrange the same type of fruits in one place. Here, the previous work is called **training data** in Data Mining terminology. So, it learns things from the training data. This is because it has a response variable that says y that if some fruit has so and so features then it is grape, and similarly for every fruit. This type of information is deciphered from the data that is used to train the model. This type of learning is called **Supervised Learning**. Such problems are listed under classical *Classification Tasks*.

**Unsupervised Learning**

[Unsupervised learning](https://www.geeksforgeeks.org/ml-types-learning-part-2/) is a type of machine learning where the algorithm is given input data without explicit instructions on what to do with it. In unsupervised learning, the algorithm tries to find patterns, structures, or relationships in the data without the guidance of labelled output.

The main goal of unsupervised learning is often to explore the inherent structure within a set of data points. This can involve identifying clusters of similar data points, detecting outliers, reducing the dimensionality of the data, or discovering patterns and associations.

There are several common types of unsupervised learning techniques:

1. [**Clustering**](https://www.geeksforgeeks.org/clustering-in-machine-learning/)**:** Clustering algorithms aim to group similar data points into clusters based on some similarity metric. K-means clustering and hierarchical clustering are examples of unsupervised clustering techniques.
2. [**Dimensionality Reduction**](https://www.geeksforgeeks.org/dimensionality-reduction/)**:** These techniques aim to reduce the number of features (or dimensions) in the data while preserving its essential information. Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are examples of dimensionality reduction methods.
3. [**Association**](https://www.geeksforgeeks.org/association-rule/)**:** Association rule learning is used to discover interesting relationships or associations between variables in large datasets. The Apriori algorithm is a well-known example used for association rule learning.

**Why Unsupervised Learning?**

The main aim of Unsupervised learning is to model the distribution of the data to learn more about the data. It is called unsupervised learning because there is no correct answer and there is no such teacher(unlike supervised learning). Algorithms are left to their own devices to discover and present an interesting structure in the data.

**Unsupervised Learning example**

Again, Suppose there is a basket and it is filled with some fresh fruits. The task is to arrange the same type of fruits in one place. This time there is no information about those fruits beforehand, it’s the first time that the fruits are being seen or discovered So how to group similar fruits without any prior knowledge about them? First, any physical characteristic of a particular fruit is selected. Suppose *colour*. Then the fruits are arranged based on the color. 

The groups will be something as shown below:

* **RED COLOR GROUP**: apples & cherry fruits.
* **GREEN COLOR GROUP**: bananas & grapes. So now, take another physical character say, *size*, so now the groups will be something like this.
* **RED COLOR** AND **BIG SIZE**: apple.
* **RED COLOR** AND **SMALL SIZE**: cherry fruits.
* **GREEN COLOR** AND **BIG SIZE**: bananas.
* **GREEN COLOR** AND **SMALL SIZE**: grapes.

The job is done! Here, there is no need to know or learn anything beforehand. That means, no train data and no response variable. This type of learning is known as Unsupervised Learning.

**Difference between Supervised and Unsupervised Learning**

The distinction between supervised and unsupervised learning depends on whether the learning algorithm uses pattern-class information. Supervised learning assumes the availability of a teacher or supervisor who classifies the training examples, whereas unsupervised learning must identify the pattern-class information as a part of the learning process.

Supervised learning algorithms utilize the information on the class membership of each training instance. This information allows supervised learning algorithms to detect pattern misclassifications as feedback to themselves. In unsupervised learning algorithms, unlabeled instances are used. They blindly or heuristically process them. Unsupervised learning algorithms often have less computational complexity and less accuracy than supervised learning algorithms.

|  | **Supervised Learning** | **Unsupervised Learning** |
| --- | --- | --- |
| **Input Data** | Uses Known and Labeled Data as input | Uses Unknown Data as input |
| **Computational Complexity** | Less Computational Complexity | More Computational Complex |
| **Real-Time** | Uses off-line analysis | Uses Real-Time Analysis of Data |
| **Number of Classes** | The number of Classes is known | The number of Classes is not known |
| **Accuracy of Results** | Accurate and Reliable Results | Moderate Accurate and Reliable Results |
| **Output data** | The desired output is given. | The desired, output is not given. |
| **Model** | In supervised learning it is not possible to learn larger and more complex models than in unsupervised learning | In unsupervised learning it is possible to learn larger and more complex models than in supervised learning |
| **Training data** | In supervised learning training data is used to infer model | In unsupervised learning training data is not used. |
| **Another name** | Supervised learning is also called classification. | Unsupervised learning is also called clustering. |
| **Test of model** | We can test our model. | We can not test our model. |
| **Example** | Optical Character Recognition | Find a face in an image. |

1. Mention a few unsupervised learning applications.

**Applications of Unsupervised Learning**

Unsupervised learning finds applications across various domains. Some notable applications include:

1. **Customer Segmentation:** Unsupervised learning algorithms can group customers based on their purchasing behavior, allowing businesses to tailor marketing strategies.
2. **Anomaly Detection:** By identifying abnormal patterns or outliers, unsupervised learning can help detect fraud, network intrusions, or manufacturing defects.
3. **Image and Text Clustering:** Unsupervised learning can automatically group similar images or texts, aiding in tasks like image organization, document clustering, or content recommendation.
4. **Genome Analysis:** Unsupervised learning algorithms can analyze genetic data to identify patterns and relationships, leading to insights in personalized medicine and genetic research.
5. Social Network Analysis: Unsupervised learning can be used to identify communities or influential individuals within social networks, enabling targeted marketing or detecting online communities.

3. What are the three main types of clustering methods? Briefly describe the characteristics of each.

Types of Clustering Methods

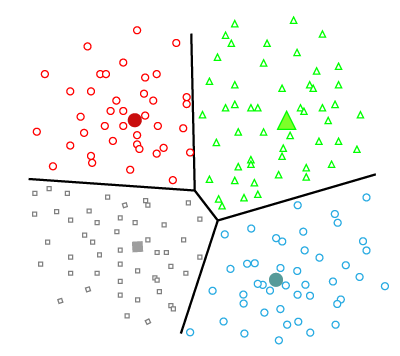
The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also). But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

Partitioning Clustering

It is a type of clustering that divides the data into non-hierarchical groups. It is also known as the **centroid-based method**. The most common example of partitioning clustering is the [**K-Means Clustering algorithm**](https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning).

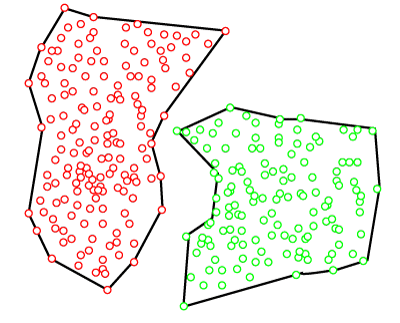
In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



Density-Based Clustering

The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected. This algorithm does it by identifying different clusters in the dataset and connects the areas of high densities into clusters. The dense areas in data space are divided from each other by sparser areas.

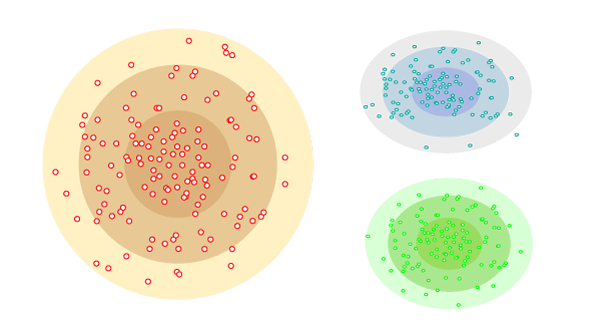
These algorithms can face difficulty in clustering the data points if the dataset has varying densities and high dimensions.



Distribution Model-Based Clustering

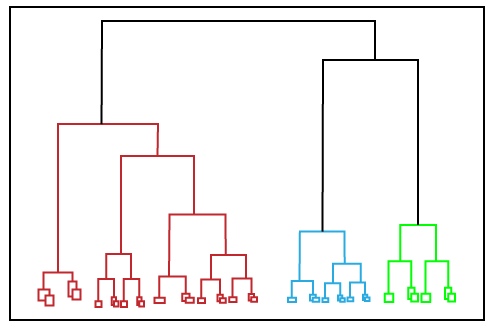
In the distribution model-based clustering method, the data is divided based on the probability of how a dataset belongs to a particular distribution. The grouping is done by assuming some distributions commonly **Gaussian Distribution**.

The example of this type is the **Expectation-Maximization Clustering algorithm** that uses Gaussian Mixture Models (GMM).



Hierarchical Clustering

Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created. In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a **dendrogram**. The observations or any number of clusters can be selected by cutting the tree at the correct level. The most common example of this method is the **Agglomerative Hierarchical algorithm**.



Fuzzy Clustering

[Fuzzy](https://www.javatpoint.com/fuzzy-logic) clustering is a type of soft method in which a data object may belong to more than one group or cluster. Each dataset has a set of membership coefficients, which depend on the degree of membership to be in a cluster. **Fuzzy C-means algorithm** is the example of this type of clustering; it is sometimes also known as the Fuzzy k-means algorithm.

4. Explain how the k-means algorithm determines the consistency of clustering.

The elbow method runs k-means clustering (kmeans number of clusters) on the dataset for a range of values of k (say 1 to 10) In the elbow method, we plot mean distance and look for the [elbow point](https://blogs.oracle.com/ai-and-datascience/post/introduction-to-k-means-clustering)where the rate of decrease shifts. For each k, calculate the total within-cluster sum of squares (WSS). This elbow point can be used to determine K.

* Perform K-means clustering with all these different values of K. For each of the K values, we calculate average distances to the centroid across all data points.
* Plot these points and find the point where the average distance from the centroid falls suddenly (“Elbow”).

5. With a simple illustration, explain the key difference between the k-means and k-medoids algorithms.

|  |  |
| --- | --- |
| K-means | K-medoids |
| K-means takes the mean of data points to create new points called centroids. | K-medoids uses points from the data to serve as points called medoids. |
| Centroids are new points previously not found in the data. | Medoids are existing points from the data. |
| K-means can only by used for numerical data. | K-medoids can be used for both numerical and categorical data. |
| K-means focuses on reducing the sum of squared distances, also known as the sum of squared error (SSE). | K-medoids focuses on reducing the dissimilarities between clusters of data from the dataset. |
| K-means uses Euclidean distance. | K-medoids uses Manhattan distance. |
| K-means is not sensitive to outliers within the data. | K-medoids is outlier resistant and can reduce the effect of outliers. |
| K-means does not cater to noise in the data. | K-medoids effectively reduces the noise in the data. |
| K-means is less costly to implement. | K-medoids is more costly to implement. |
| K-means is faster. | K-medoids is comparatively not as fast. |

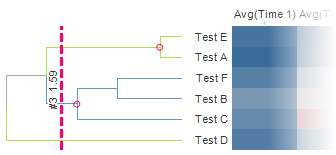
6. What is a dendrogram, and how does it work? Explain how to do it.

# Dendrograms and Clustering

A dendrogram is a tree-structured graph used in heat maps to visualize the result of a hierarchical clustering calculation. The result of a clustering is presented either as the distance or the similarity between the clustered rows or columns depending on the selected distance measure. See [Distance Measures Overview](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/hc/hc_distance_measures_overview.htm) and the detailed description for each measure for further information about the available distance measures. You can perform hierarchical clustering on an existing heat map by opening the [Dendrograms page](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/heat/heat_properties_dendrograms.htm) of the Visualization Properties. You can also use the [Hierarchical Clustering tool](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/hc/hc_what_is_the_hierarchical_clustering_tool.htm) to cluster with a data table as the input. To learn more about hierarchical clustering and heat maps, see [Overview of Hierarchical Clustering Theory](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/hc/hc_method_overview.htm) and [What is a Heat Map?](https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/heat/heat_what_is_a_heat_map.htm) respectively. Note that only numeric columns will be included when clustering.

**Row Dendrograms**

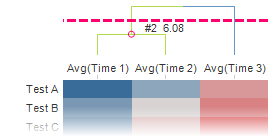
The row dendrogram shows the distance or similarity between rows and which nodes each row belongs to, as a result of clustering. An example of a row dendrogram is shown below.



The individual rows in the clustered data are represented by the right-most nodes, the leaf nodes, in the row dendrogram. Each node in the dendrogram represents a cluster of all rows that lie to the right of it in the dendrogram. The left-most node in the dendrogram is therefore a cluster that contains all rows. The vertical dotted line is the pruning line, which can be dragged sideways in the dendrogram. The values next to the pruning line indicate the number of clusters starting from the current position of the line, as well as the calculated distance or similarity at that position. In the example above, the calculated distance is 1.59, and there are three clusters starting at the position of the pruning line. The upper two, indicated by pink circles, contain two or more rows, while the lower cluster contains only one individual row.

**Column Dendrograms**

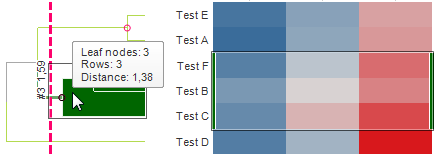
The column dendrogram is drawn in the same way as the row dendrogram, but shows the distance or similarity between the variables (the cell value columns).



At the position of the pruning line in the above example, there are two clusters. The left-most cluster contains two columns, while the right-most cluster contains only one individual column. The calculated distance is 6.08.

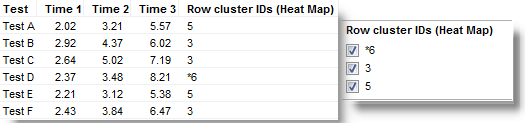
**Interacting with the dendrogram**

The dendrogram makes it easy to highlight and mark in the heat map. You can mouseover the dendrogram to highlight clusters and their corresponding cells in the heat map. You can click to mark a cluster. This will also mark the corresponding cells in the heat map, as in the example below. The tooltip displays information about the cluster.



**Clustering**

As mentioned, a dendrogram is added to the heat map when clustering is performed. A new column is also added to the data table, and made available in the filters panel. The cluster column is dynamic, and the position of the pruning line decides its content. The example below shows what the cluster column and cluster filter would look like for the row dendrogram above.



The cluster column contains unique identifiers for the cluster nodes corresponding to the position of the pruning line. In the example above, two cluster nodes are identified. Test B, Test C, and Test F belong to the cluster node with identifier 3, while Test A and Test E belong to the cluster node with identifier 5. The third identifier, \*6, is a leaf node, containing Test D. The cluster column makes it possible to filter out entire clusters at a time. You can also use it to color or trellis other visualizations by.

7. What exactly is SSE? What role does it play in the k-means algorithm?

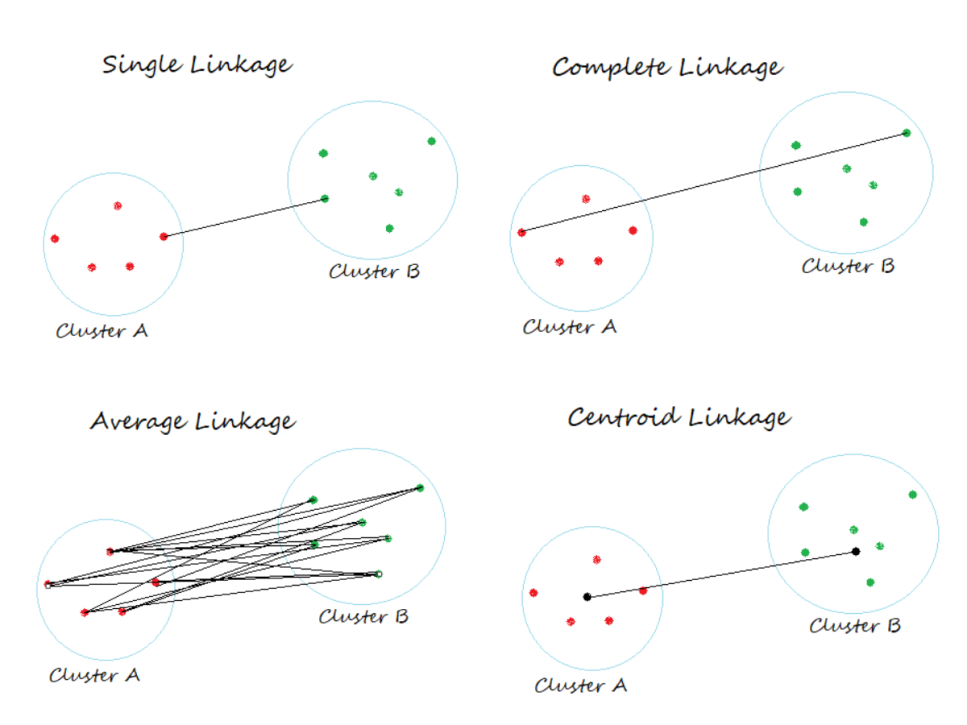
The Sum of Squared Errors (SSE) measures the sum of the squared distances between each data point and its assigned centroid. A lower SSE value indicates that the data points within each cluster are closer to their respective centroids, suggesting a better clustering solution. However, SSE alone may not provide a complete understanding of the clustering quality and should be used in conjunction with other evaluation methods.

8. With a step-by-step algorithm, explain the k-means procedure.

1. Decide how many clusters you want, i.e. choose k
2. Randomly assign a centroid to each of the k clusters
3. Calculate the distance of all observation to each of the k centroids
4. Assign observations to the closest centroid
5. Find the new location of the centroid by taking the mean of all the observations in each cluster
6. Repeat steps 3-5 until the centroids do not change position

9. In the sense of hierarchical clustering, define the terms single link and complete link.

* Furthermore, in Single Linkage Clustering, the distance between two clusters is the minimum distance between members of the two clusters.
* Additionally, in Complete Linkage, the distance between two clusters is the maximum distance between members of the two clusters.



10. How does the apriori concept aid in the reduction of measurement overhead in a business basket analysis? Give an example to demonstrate your point.

# Apriori Algorithm

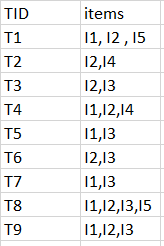
Prerequisite – [Frequent Item set in Data set (Association Rule Mining)](https://www.geeksforgeeks.org/frequent-item-set-in-data-set-association-rule-mining/)  
**Apriori algorithm** is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.

To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called *Apriori property* which helps by reducing the search space.

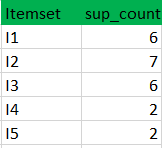
**Apriori Property –**  
All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes that

*All subsets of a frequent itemset must be frequent(Apriori property).  
If an itemset is infrequent, all its supersets will be infrequent.*

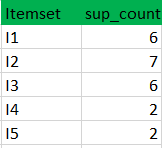
Before we start understanding the algorithm, go through some definitions which are explained in my previous post.  
Consider the following dataset and we will find frequent itemsets and generate association rules for them.

  
minimum support count is 2  
minimum confidence is 60%

**Step-1:**K=1  
(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**

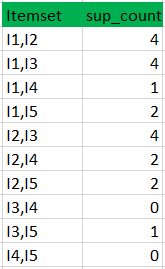


(II) compare candidate set item’s support count with minimum support count(here min\_support=2 if support\_count of candidate set items is less than min\_support then remove those items). This gives us itemset L1.

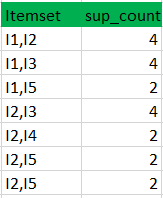


**Step-2:**K=2

* Generate candidate set C2 using L1 (this is called join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common.
* Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
* Now find support count of these itemsets by searching in dataset.

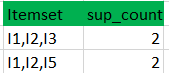


(II) compare candidate (C2) support count with minimum support count(here min\_support=2 if support\_count of candidate set item is less than min\_support then remove those items) this gives us itemset L2.

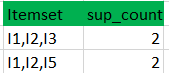


**Step-3:**

* + Generate candidate set C3 using L2 (join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common. So here, for L2, first element should match.  
    So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5}
  + Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
  + find support count of these remaining itemset by searching in dataset.



(II) Compare candidate (C3) support count with minimum support count(here min\_support=2 if support\_count of candidate set item is less than min\_support then remove those items) this gives us itemset L3.



**Step-4:**

* + Generate candidate set C4 using L3 (join step). Condition of joining Lk-1 and Lk-1 (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.
  + Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
  + We stop here because no frequent itemsets are found further