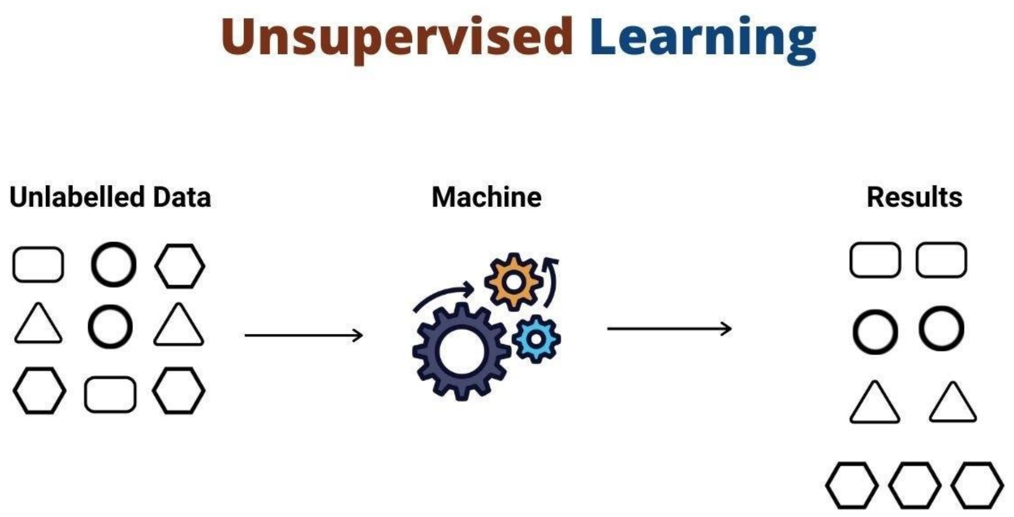


Week 7: Unsupervised Learning

Introduction to Unsupervised Learning

Overview:

- Unsupervised learning deals with unlabeled data.

- Unlike supervised learning, there are no target variables or labeled outcomes.

- The algorithm tries to find patterns, relationships, or structures within the data.

Key Points

No Labeled Outcomes:

In unsupervised learning, we don't have labeled target variables or outcomes to guide the algorithm.

The goal is to uncover hidden patterns without predefined answers.

Discovering Patterns:

The algorithm autonomously identifies relationships, structures, or groupings within the data.

This makes it particularly useful for exploring datasets where we have little to no prior knowledge.

Common Applications:

Unsupervised learning is applied in various domains, such as customer segmentation, anomaly detection, and exploratory data analysis.

Use Cases of Unsupervised Learning

Customer Segmentation

- Customer Segmentation involves grouping customers based on shared characteristics, behaviors, or preferences.

Application:

E-commerce platforms can use clustering algorithms to identify customer segments for targeted marketing strategies.

Benefits:

Tailored marketing campaigns.

Improved customer experience.

Use Cases of Unsupervised Learning

Anomaly Detection

- Anomaly Detection identifies unusual patterns or outliers in a dataset.

Application:

Cybersecurity applications use anomaly detection to identify unusual network activity.

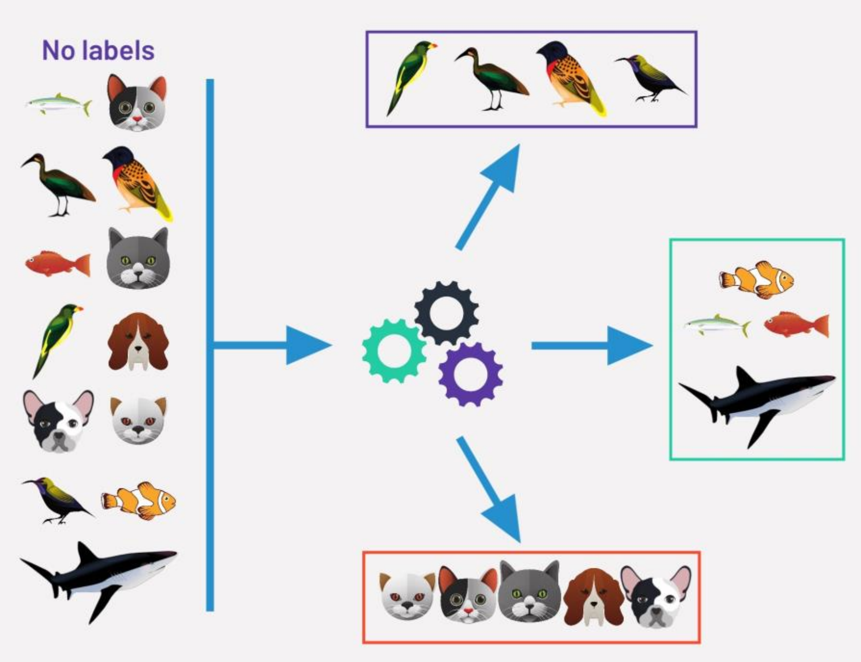
Benefits:

Early detection of fraudulent activities.

Improved system security.

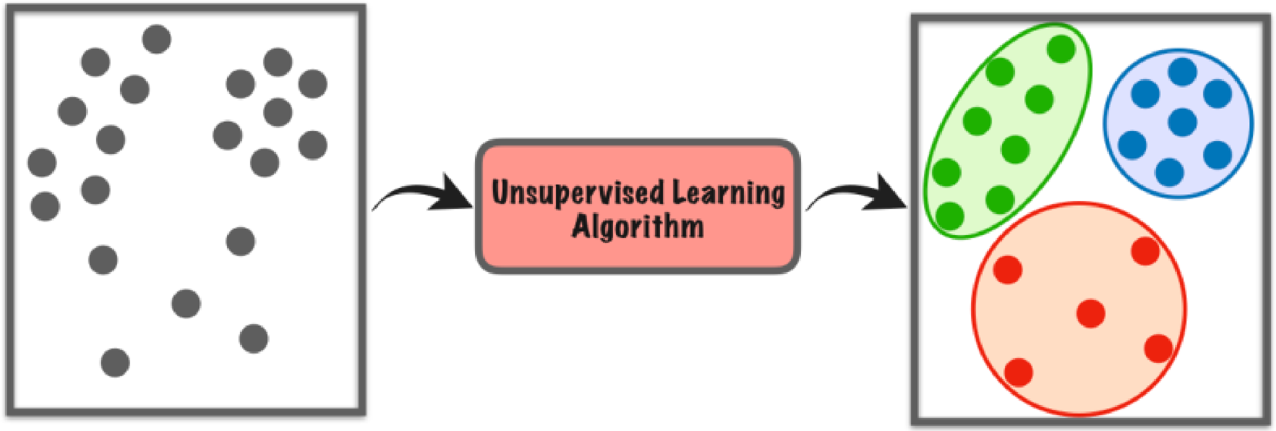
Analogical Explanation

- From the example below, we see how a machine can learn to find patterns in unlabeled data.
- Though this may be obvious when it comes to land, water, and air animals, it could be much less obvious when dealing with massive datasets.



Visual Representation

A simple visual representation: Unlabeled data points without predefined categories.



Types of Unsupervised Learning

Clustering:

- Clustering algorithms group similar data points together, revealing inherent structures within the dataset.
- Examples: K-Means, Hierarchical Clustering, DBSCAN.

Association:

- Association algorithms identify relationships and patterns within the data, unveiling rules that describe large portions of the dataset.
- Example: Apriori algorithm for market basket analysis.

Dimensionality Reduction:

- Reducing the number of features while preserving information.
- Examples: PCA (Principal Component Analysis), t-SNE (t-Distributed Stochastic Neighbor Embedding).

Real-World Applications

Clustering: Customer segmentation for targeted marketing.

Association: Analyzing patterns in transaction data for market basket analysis.

Dimensionality Reduction: Reducing features for image processing or speech recognition.

Clustering

K-Means Clustering:

- Divides data into 'k' clusters based on similarity.
- Steps: Initialization, Assignment, Update.
- Considerations: Choosing the right 'k,' handling outliers.

Hierarchical Clustering:

- Creates a tree of clusters.
- Agglomerative (bottom-up) or divisive (top-down) approaches.
- Visualization using dendrogram.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

- Groups together data points that are close to each other.
- Recognizes noise/outliers.

K-Means Clustering

Definition: Divides data into '*k*' clusters based on similarity.

Steps:

1. **Initialization**: Randomly places 'k' cluster centroids.

2. **Assignment**: Assigns each data point to the nearest centroid.

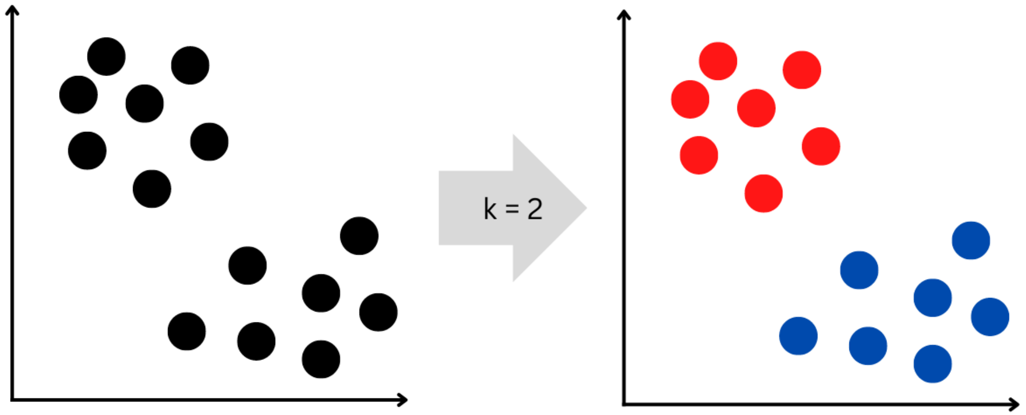
3. **Update**: Recalculates centroids based on the assigned points.

K-Means Clustering

Considerations:

Choosing 'k’: Selecting the right number of clusters is crucial.

Handling Outliers: K-Means can be sensitive to outliers, affecting cluster centroids.



K-Means Clustering Algorithm

Hierarchical Clustering

- Hierarchical Clustering is a method of grouping similar data points into clusters, forming a tree-like structure or hierarchy.

Agglomerative (Bottom-Up):

- Starts with individual data points and progressively merges them into clusters.
- The hierarchy is built from the bottom up.

Divisive (Top-Down):

- Starts with one big cluster containing all data points and recursively divides it into smaller clusters.
- The hierarchy is built from the top down.

Visual Representation

Dendrogram:

- The result of hierarchical clustering is often represented by a dendrogram.
- A dendrogram is a tree-like diagram that illustrates the hierarchy of clusters.

When is Hierarchical Clustering Useful?

- Useful when the goal is to understand relationships between data points in a hierarchical manner.
- Commonly used in biological taxonomy, customer segmentation, and more.

Agglomerative Hierarchical Clustering

- It starts by considering each observation as a singleton cluster (cluster with only one data point).

- Then iteratively merges clusters until only one cluster is obtained.

- This process is also known as the bottom-up approach.

Example

- Imagine organizing different animals based on similarities in their features:

Step 1: Each animal is its own cluster.

Step 2: Merge animals that share common features.

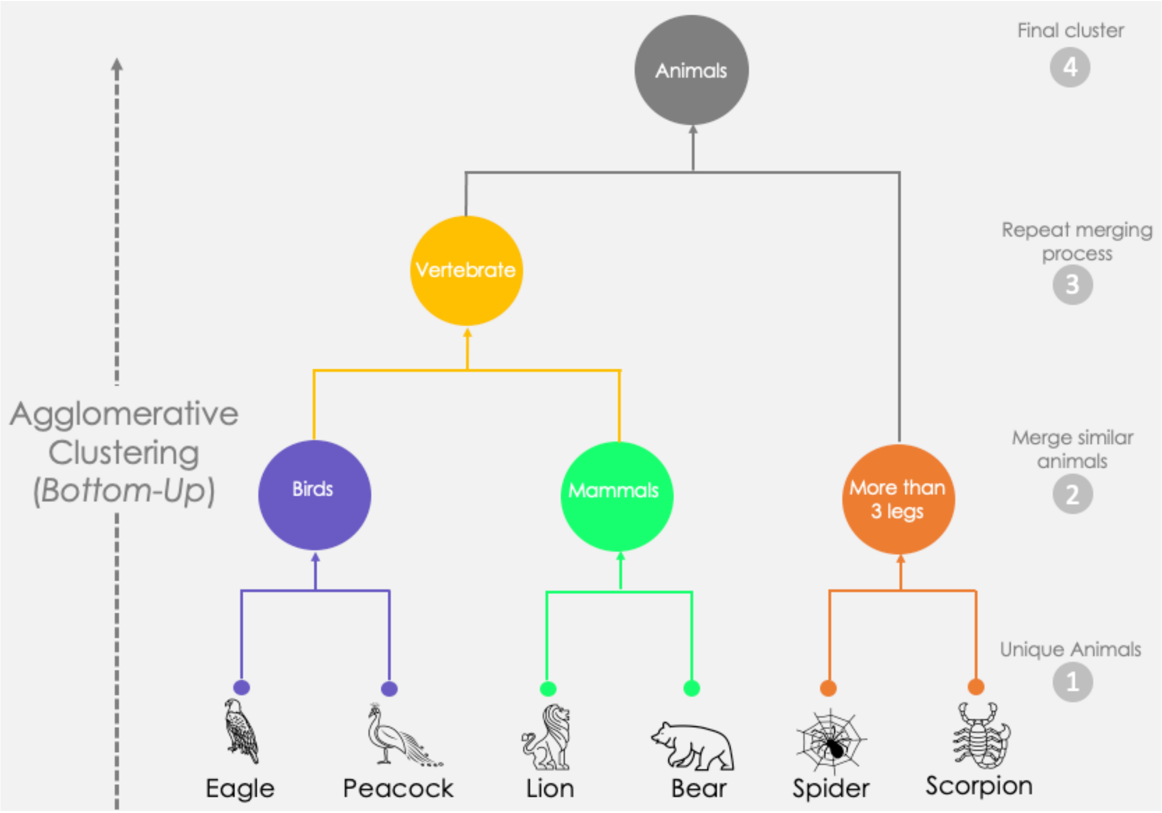
Step 3: Continue merging until all animals are in one big cluster.

Agglomerative Hierarchical Clustering

- We start by considering each animal to be its unique cluster.
- Then we generate three different clusters from those unique animals based on their similarities:
  - **Birds**: Eagle and Peacock
  - **Mammals**: Lion and Bear
  - **More than three leg animals**: Spider and Scorpion.
- We repeat the merging process to create the vertebrate cluster by combining the two most similar clusters: Birds and Mammals.
- After this step, the remaining two clusters, Vertebrate and More than three legs, are merged to create a single Animals cluster.

Agglomerative Hierarchical Clustering

Dendrogram of Agglomerative Clustering Approach



Divisive Hierarchical Clustering

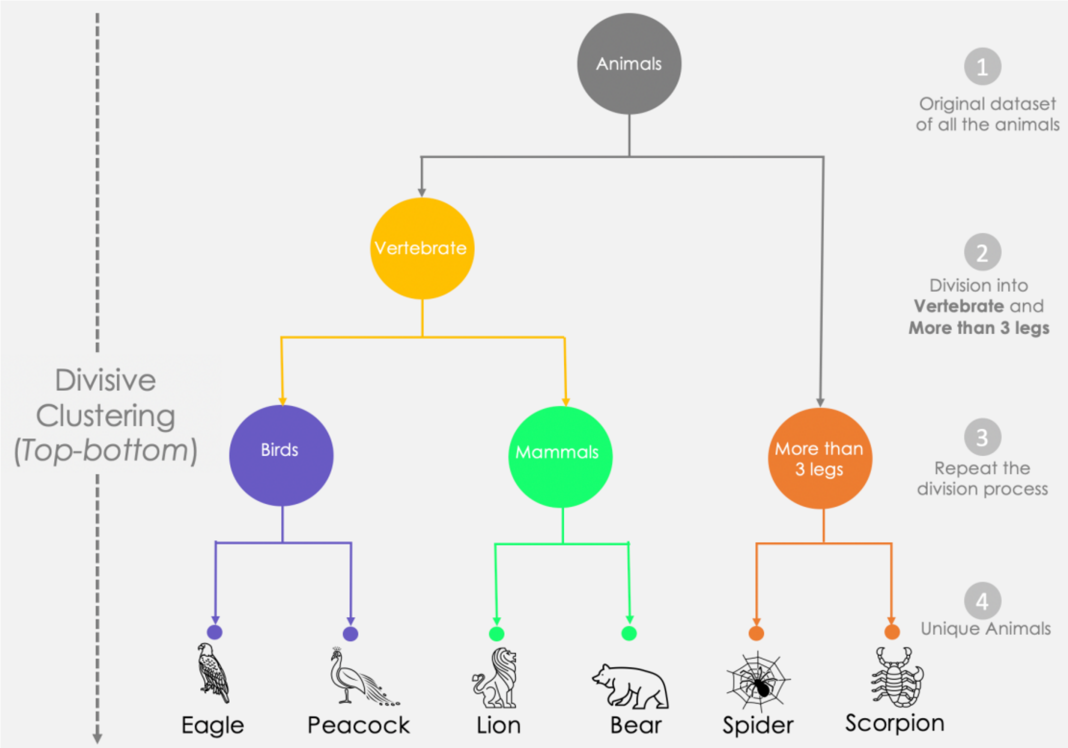
- On the other hand, divisive clustering is top-down because it starts by considering all the data points as a unique cluster. Then it separates them until all the data points are unique.

From this divisive approach graphic:

- We notice that the whole animal dataset is considered as a single bloc.
- Then, we divide that block into two clusters: Vertebrate and More than 3 legs.
- The division process is iteratively applied to the previously created clusters until we get unique animals.

Divisive Hierarchical Clustering

Dendrogram of Divisive Clustering Approach



DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- DBSCAN, is a clustering algorithm that groups data points based on their density in the feature space.
- It calculates density in the feature space using two parameters: **epsilon (ε or eps)** and **min_samples**.

Epsilon (ε or eps): This parameter defines the radius around a data point within which the algorithm counts the number of other data points. It represents the maximum distance between two samples for one to be considered as in the neighborhood of the other.

Min_samples: This parameter specifies the minimum number of data points required to form a dense region.

DBSCAN

- The density calculation process:

Core Point: A data point is considered a core point if there are at least "min_samples" data points (including itself) within its epsilon neighborhood.

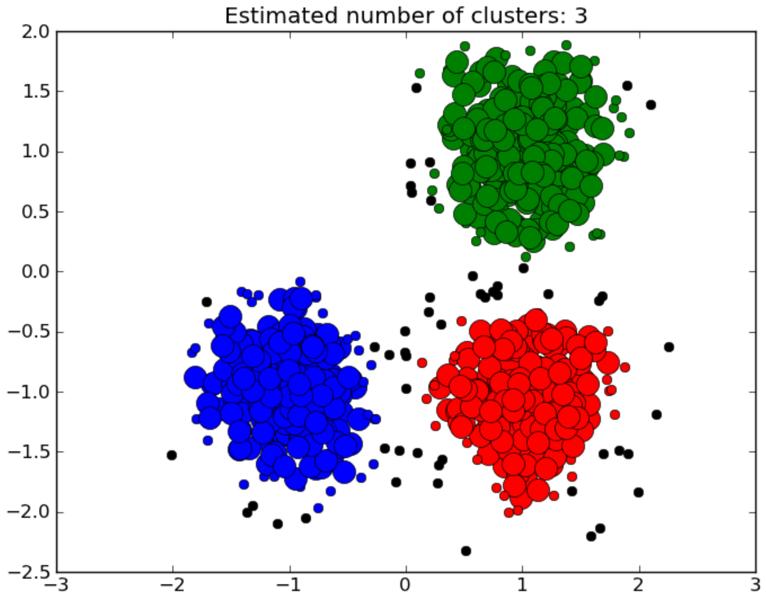
Border Point: A data point is considered a border point if it is within the epsilon neighborhood of a core point but does not have enough neighbors to be a core point itself.

Noise (or Outlier) Point: A data point is considered a noise (or outlier) point if it is not a core point and does not have enough neighbors to be a border point.

Example

A visual representation of how DBSCAN forms clusters based on the density of data points.

- Shows how clusters are formed around core points and extend to include border points.
- Illustrates isolated points that do not meet the density criteria and are considered noise.



When is DBSCAN Useful?

Flexible Clustering:

- Suitable for datasets with irregular shapes and varying cluster densities.
- Effective in identifying clusters of different shapes and sizes.

Noise Handling:

- Robust to outliers and able to identify noise points.

Example

Imagine grouping GPS coordinates of delivery locations:

- **Core Points**: Locations with many nearby delivery points.
- **Clusters**: Areas with high delivery density.
- **Noise**: Isolated delivery points.

Association Rule Mining

Association in unsupervised learning refers to discovering relationships or patterns among items in a dataset.

How Does it Work?

1. Frequent Itemsets:
   - Association algorithms identify sets of items that frequently co-occur in the dataset.
   - These sets are called frequent itemsets.

2. Association Rules:
   - From frequent itemsets, rules are generated to express relationships between items.
   - Rules typically have two parts: the antecedent (items on the left) and the consequent (items on the right).

3. Metrics:

Association rules are evaluated using metrics like support, confidence, and lift.

Association Rule Mining

- This type of unsupervised machine learning takes a rule-based approach to discovering interesting relationships between features in a given dataset. It works by using a measure of interest to identify strong rules found within a dataset.

- We typically see association rule mining used for market basket analysis: this is a data mining technique retailers use to gain a better understanding of customer purchasing patterns based on the relationships between various products.

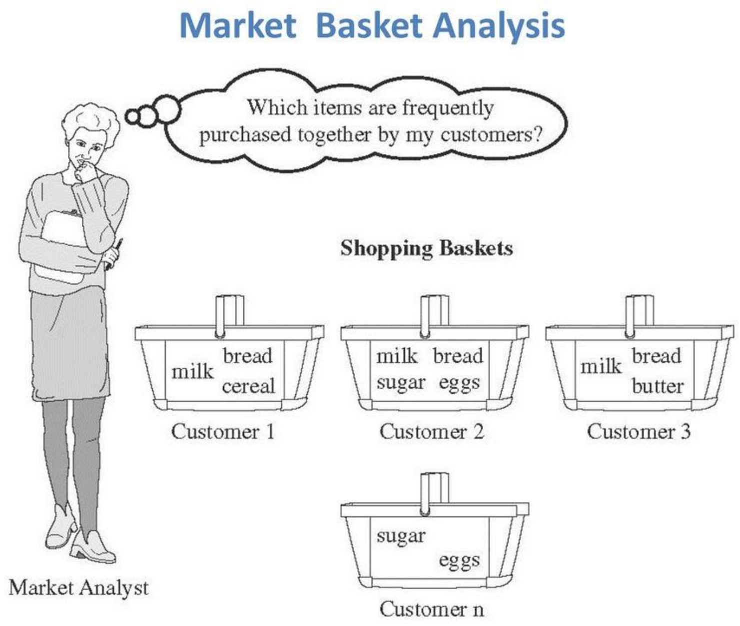
- The most widely used algorithm for association rule learning is the Apriori algorithm. However, other algorithms are used for this type of unsupervised learning, such as the Eclat and FP-growth algorithms.

Example

Consider a supermarket transaction dataset:

Frequent Itemsets: Frequent pairs of items purchased together, like {Milk, Bread}.

Association Rule: If a customer buys Milk, there's a high confidence they'll also buy Bread.



Apriori Algorithm

The Apriori algorithm is a classic association rule mining algorithm used to discover frequent itemsets in a dataset and derive association rules.

How Does it Work?

Frequent Itemsets:

- Apriori identifies sets of items (itemsets) that frequently co-occur in a dataset.
- Itemsets are considered frequent if they meet a predefined support threshold.

Join and Prune:

- Apriori employs a "join and prune" strategy to generate candidate itemsets efficiently.
- It starts with individual items, then joins and prunes itemsets based on their support.

Association Rules:

- From frequent itemsets, association rules are derived.
- Rules express relationships between items, typically having an antecedent (left side) and a consequent (right side).

Key Terms

1. **Support**:
   - Measures the frequency of an itemset in the dataset.
   - Calculated as the number of transactions containing the itemset divided by the total number of transactions.
   - High support indicates that the itemset is frequent.

$$\text{Support}(X) = \frac{\text{Transactions containing } X}{\text{Total transactions}}$$

Key Terms

2. **Confidence**:

- Confidence measures how likely item Y is purchased when item X is purchased, indicating the strength of the association between X and Y.
- Indicates the likelihood of a rule being true.
- Calculated as the support of the combined itemset divided by the support of the antecedent (the items on the left side of the rule).
- High confidence implies a strong association between the antecedent and consequent.
- High confidence means that the rule is accurate or trustworthy.

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

Key Terms

3. **Lift**:

- Lift measures how much more likely item Y is purchased when item X is purchased compared to when Y is purchased independently of X.
- Compares the likelihood of the rule with and without the antecedent.
- Lift is a measure of the strength of association between two items in an association rule.
- Lift > 1 indicates that the antecedent and consequent are associated more than expected.
- Lift = 1 suggests that the antecedent and consequent are independent.
- Lift < 1 indicates that the antecedent and consequent are associated less than expected.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{\text{Support}(Y)}$$

Example

- Widely used in market basket analysis to understand customer purchasing patterns.
- Consider a dataset of customer transactions in a supermarket.

Frequent Itemsets: {Milk, Bread}, {Eggs, Cheese}, etc.

Association Rule: If a customer buys Milk and Bread, there's a high confidence they'll also buy Eggs.

ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
…	…

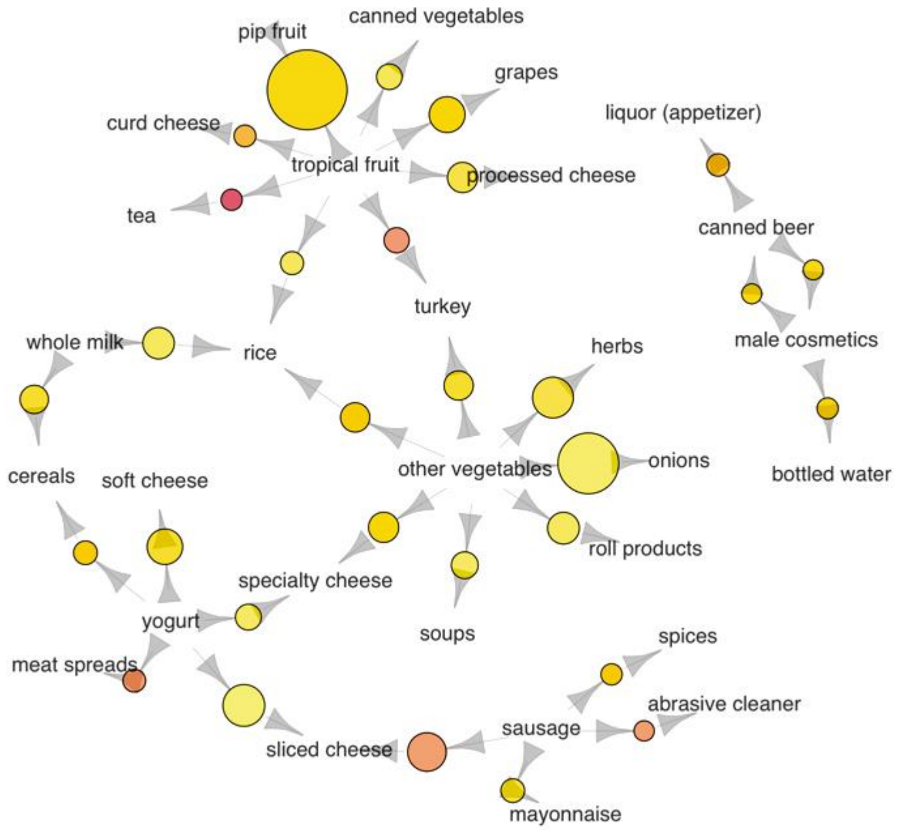
market basket transactions

{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule

Visual Representation

Graphical representation



Challenges in Unsupervised Learning

Lack of Labeled Data for Validation:

- Without labeled data, it's challenging to validate the performance of unsupervised learning models.
- Evaluation metrics may be less straightforward compared to supervised learning.

Determining the Optimal Number of Clusters:

- In clustering algorithms, determining the right number of clusters (k) is often subjective.
- Techniques like the **elbow method** or **silhouette analysis** can assist, but interpretation is required.