**1. Let min\_sup = a % , min\_conf = b %. Using Apriori**

\* With min\_sup = a % and the number of samples = c

=> min\_sup\_count =

+ We have candidate 1-itemsets and their support as C1:

|  |  |
| --- | --- |
| Itemset | sup |
| … | … |

=> Frequent 1-itemsets satisfy min\_sup\_count as L1:

+ We have candidate 2-itemsets and their support as

C2 = L1 ⋈ L1: ( kẻ bảng )

=> Frequent 2-itemsets satisfy min\_sup\_count as L2: …

**Stop condition:**

1. Because L2 has only one itemset, we don’t need to find candidate 3-itemsets. So all frequent itemset:

2. Because we can’t find frequent 2-itemsets satisfying min\_sup\_count, so all frequent itemset: L = {{ }, { } }

\* Consider maximal frequent itemset { 3 } : (sup) has the non-empty subsets: { 1 } : (sup) , { 2 } : (sup)

|  |  |
| --- | --- |
| association rule | confidence |
| { 1 } 🡪 { 2 } | p(maximal) / p({1} xảy ra) |
| { 2 } 🡪 { 1 } |  |

With min\_conf = b % , we find all association rules:

**2. Let min\_sup = a % , min\_conf = b %. Using FP-g**

We have database: ( kẻ bảng )

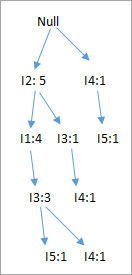
**+ Step1: Build FP Tree**

Frequent 1-itemsets satisfy **min\_sup\_count** and **sorted** in descending order as T: ( kẻ bảng )

From T, Ordered frequent items for database:

|  |  |
| --- | --- |
| ID | Item |
| 1 | *(Đưa các giá trị có sup cao hơn lên đầu dựa trên bảng T)* |
| … | *(Note: dòng nào có giá trị mà không thỏa min\_sup trong T thì bỏ)* |

**A screenshot of a pattern

Description automatically generated with low confidence**

**Step2: Conditional Pattern Base**

**+ Step3: Conditional FP-Tree (min\_sup\_count)**

**A screenshot of a computer

Description automatically generated with low confidence**

**+ Step4: Frequent Patterns Generated**

**A screenshot of a computer

Description automatically generated with low confidence**

**2. Rough set**

INDIS(B) = { *Tìm tập các phần tử có giá trị giống nhau trên cùng tập thuộc tính của toàn dataset* }

Lower approximation: BX = *tập phần tử thuộc X*.

Upper approximation: X *= chỉ cần 1 phần tử trong tập có trong X*.

Boundary region: BR = = *phần tử có trong upper mà không có trong lower*.

Outsider: B\_Out = *tập có trong INDIS(B) mà không có trong X*. /// **Quality Coefficient**:

If =1 : X is clear approximation regarding to B.

If <1 : X is rough approximation regarding to B.

**3. Determine the root**

**\*Gini:**

+ (Attribute) has … possible values : A, B and C.

= x + …

The attribute “ X ” with the smallest Gini is selected as the root node of Decision tree.

**\*Information gain:**

The data S has k samples with m samples belong to Yes, and n samples belong to class No. Entropy of S:

+ (Attribute) has … possible values : A, B and C.

The attribute “ X ” with the largest information gain is selected as the root node of Decision tree.

**4. Naïve Bayesian using Laplacican correction**

+ (Thuộc tính qđ) = Yes

p(qđ = Yes) =

p1= p(X1 = | qđ = Yes) =

p(qđ = Yes) x p(X | qđ = Yes) = p(qđ = Yes) x p1 x p..

+ (Thuộc tính qđ) = No (*làm tương tự*)

Because p(qđ = Yes | X ) > p(qđ = No | X ) => X …

**5. Kmeans**

Compute centroid vector v1(v11, v12), v2(v21, v22), v3 …

v1 = ( , )

Calculate distances using Euclide distance:

CT: d (P1, v1) =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | v1 | v2 | v3 | note |
| P1 |  |  |  | P1 belongs to c… |

Update matrix M: ( kẻ ma trận M1)

Loop until M does not change.

**Discrete Attribute:** Has only a finite or countably infinite set of values. Sometimes, represented as integer variables. Binary attributes are a special case of discrete attributes. (zip codes).

**Continuous Attribute:** Has real numbers as attribute values.

Practically, real values can only be measured and represented using a finite number of digits. Continuous attributes are typically represented as floating-point variables. (temp, height, weight).

Dataset are made up of **data objects**. A data object represents an entity. Ex: university database: students, professors, courses.

Data objects are described by attributes.

Database rows 🡪 data objects; columns 🡪 attributes.

**Attribute**: a data field, representing a characteristic or feature of a data object.

**Type Attribute:** - Nominal: categories, states, or “names of things”. Hair\_color = {auburn, black, blond, brown, grey, red}

- Binary: Nominal attribute with only 2 states (0 - 1)

+ Symmetric binary: both outcomes equally important (gender)

+ Asymmetric binary: not equally important (medical test: + -)

+ Ordinal: Values have a meaningful order (ranking) but magnitude between successive values is not known.

Size = {small, medium, large}, grades, army rankings.

Numeric: + Quantity (integer or real-valued)

+ Interval: Measured on a scale of equal-sized units.Values have order. No true zero-point(temp in C˚or F˚, calendar dates). + Ratio: Inherent zero-point. We can speak of values as being an order of magnitude larger than the unit of measurement (10 K˚ is twice as high as 5 K˚). Ex:(temperature in Kelvin, monetary quantities)

**Supervised Learning** (classification): The training data (observations, measurements,…) are accompanied by labels indicating the class of the observations. New data is classified based on the training set.

**Unsupervised Learning** (Clustering): The class labels of training data is unknown. Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data.

**The main differences** of supervised vs unsupervised learning:

The need for labelled data in supervised machine learning.

The problem the model is deployed to solve. Supervised machine learning is generally used to classify data or make predictions, whereas unsupervised learning is generally used to understand relationships within datasets and hidden patterns .

Supervised machine learning is much more resource-intensive because of the need for labelled data.

In unsupervised machine learning it can be more difficult to reach adequate levels of explainability because of less human oversight.

**Applications of Clustering:** Customer segmentation in marketing. Image segmentation in computer vision. Document clustering in text mining. Anomaly detection in cybersecurity.

**Applications of Classification:** Email spam detection. Sentiment analysis in natural language processing. Object recognition in CV.

**Preprocess Data:** Measures data quality => A multidimensional view. Accuracy: correct or wrong. Completeness: not recorded, unavailable, …Consistency: some modified but some not. Timeliness: timely update? Believability: how trustable the data are correct? Interpretability: how easily the data can be understood?

**Major Tasks in Data Preprocessing:**

+ Data cleaning: Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies. + Data integration:

Integration of multiple databases, data cubes, or files

+ Data reduction: Dimensionality reduction, Numerosity reduction,

Data compression. + Data transformation and data discretization:

Normalization, Concept hierarchy generation.

**Missing data** may be due to: equipment malfunction, inconsistent with other recorded data and thus deleted, data not entered due to misunderstanding, certain data may not be considered important at the time of entry, not register history or changes

**Handle missing data**: Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably. Fill in the missing value manually: tedious + infeasible? Fill in it automatically with a global constant : e.g., “unknown”. The attribute mean for all samples belongs to the same class. The most probable value: inference-based such as Bayesian formula or decision tree.

**Noise**: random error or variance in a measured variable

Incorrect attribute values may be due to: faulty data collection instruments, data entry problems, data transmission problems, technology limitation, inconsistency in naming convention.

Handle noisy data: + Binning: first sort data and partition into (equal-frequency) bins then one can smooth by bin means, smooth by bin median, smooth by bin boundaries.

+ Regression: smooth by fitting the data into regression functions. + Clustering: detect and remove outliers. +Combined computer and human inspection: detect suspicious values and check by human.

Example: Data Mining in **Health Care**. Application: Disease Diagnosis. Data: The data used for this application may include patient health records, medical history, laboratory test results, symptoms, demographic information, and other relevant clinical data. Data Mining Method: Classification algorithms such as Decision Trees, Support Vector Machines (SVM), or Artificial Neural Networks (ANN) can be used for disease diagnosis. The goal is to develop a model that can predict the likelihood of a particular disease or condition based on the patient's data. By analyzing patterns and relationships in the data, the model can assist healthcare professionals in making accurate and timely diagnoses.

DM **in Marketing and Sales**. Application: Customer Segmentation. Data: The data used for this application may include customer demographics, purchase history, browsing behavior, social media interactions, customer feedback, and other relevant marketing data. Data Mining Method: Clustering algorithms such as K-means clustering, Hierarchical clustering, or Gaussian Mixture Models can be used for customer segmentation. The goal is to identify distinct groups or segments of customers with similar characteristics and behaviors. For example, consider a retail company that wants to segment its customer base to tailor marketing strategies and promotions. The dataset would include customer information such as age, gender, income, past purchases, website browsing patterns, and engagement with marketing campaigns.

When **preprocessing real data for data mining**, difficult problems can arise.

Data Integration and Cleaning: Real-world datasets are often collected from multiple sources. Datasets may come from different systems or sources, resulting in variations in data formats, naming conventions, or units of measurement. Real data can contain inconsistencies, errors, or duplicates, affecting the quality and reliability of the analysis.

**Dimensionality Reduction**: Real-world datasets can be high-dimensional, containing a large number of features or variables. Analyzing high-dimensional data can be computationally So, aplying dimensionality reduction techniques helps reduce computational complexity, improve model performance, and enhance interpretability.