NITUK Data Mining Class Notes

Arnab Dana, Masters of Technology.

▼ Syllabus NITUK

Data Mining :

Introduction, Data analysis like Data visualization,

probability, histograms, multinomial distributions.

Data Mining and Knowledge Discovery in Databases,

Data Mining Functionalities,

Data Pre-processing, Data Cleaning,

Data Integration and Transformation, Data Reduction,

Data Discretization and Concept Hierarchy Generation.

Overview of data mining, data mining tasks, data mining tools.

Processing and visualizing data:

Data types, Data quality, Data pre-processing, Measures of similarity,

Visualization.

Association Rule Mining:

Frequent itemset generation algorithms, Rule generation algorithms,

Compact representation, Evaluation measures.

Algorithms:

Introduction to Supervised and unsupervised classification.

Advanced Concepts:

Introduction, Sequential Pattern Mining, Mining Text and Web data,

Graph mining, Mining Spatiotemporal and Trajectory Patterns,

Multivariate Time Series (MVTS) Mining,

Complex data mining.

Applications :

Healthcare, Fraud detection, Intrusion detection, Market basket analysis,
Banking and Finance.

▼ Resource

▼ Content

Number	Content	Comments #
1	***Basic***	done
2	***Conditional Probability and Bayesian Classification***	done
3	***Unsupervised Learning > Association Rule Mining***	done
4	***Association Rule Mining> Apriori Algorithm***	
5	***Association Rule Mining > FP-growth (Frequent Pattern-growth)***	done
6	***Unsupervised Learning > Clustering***	done
7	***Clustering > K-Means, Error Calculation***	done
8	***Clustering > Nearest Neighbor Clustering***	done
9	***Clustering > Agglomerative/Hierarchical Clustering***	done
10	***Clustering > DBSCAN***	done
11	***Silhouette Coefficient (SC)***	done

Number	Content	Comments #
12	***Supervised Learning > K-NN and Weighted K-NN***	done
13	***Supervised > Classification > Decision Tree > ID3***	done
14	***Supervised > Classification > Decision Tree > C4.5***	done need to verify
15	***Supervised > Classification > Decision Tree > CART***	stuck need to verify
16	***Supervised > Rule Based classification***	done
17	***Performance Measure for Supervised Learning***	done need example
18	***Data Mining Introduction***	
19	***Applications***	
20	***Advanced Concepts***	

▼ *1. Basic*

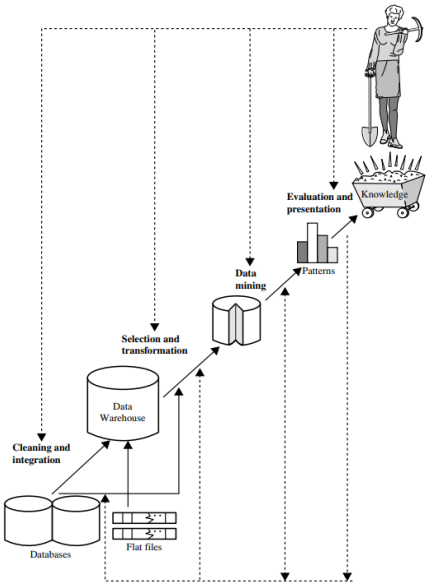
- **Definition:** Data mining is the process of extracting and analyzing patterns from large datasets. It involves the use of algorithms, statistics, and machine learning to identify hidden relationships, correlations, and anomalies in data.
- **Goal:** The goal of data mining is to transform raw data into actionable insights that can be used to improve decision-making, optimize processes, and gain a competitive edge.
- **Types of Data Mining Techniques:** There are various data mining techniques that are used to extract different types of patterns from data. Some common techniques include:
  - **Classification:** Classifies data into predefined categories.
  - **Clustering:** Groups data into clusters based on their similarities.
  - **Association rule mining:** Discovers relationships between different attributes of data.
  - **Predictive modeling:** Uses data to predict future outcomes.
- **Applications of Data Mining:** Data mining is used in a wide range of applications, including:
  - **Retail:** Customer segmentation, product recommendations, fraud detection
  - **Finance:** Risk analysis, fraud detection, credit scoring
  - **Healthcare:** Patient profiling, disease diagnosis, clinical trial analysis
  - **Marketing:** Target marketing, campaign optimization, customer engagement
  - **Telecommunications:** Customer churn prediction, network optimization
- **Challenges of Data Mining:** Data mining poses several challenges, including:
  - **Data quality:** Dealing with noisy, incomplete, or inconsistent data
  - **Scalability:** Handling massive datasets efficiently
  - **Interpretability:** Explaining the results of data mining algorithms
- **Data Mining Tools:** There are various data mining tools available, including commercial software packages and open-source tools. Some popular tools include:
  - **SAS Enterprise Miner**
  - **IBM SPSS Modeler**
  - **KNIME Analytics Platform**
  - **RapidMiner**
  - **Apache Mahout**
- **Data Mining in the Future:** Data mining is a rapidly evolving field with new techniques and applications being developed all the time. As we generate more and more data, data mining will become even more important for extracting valuable insights.

The knowledge discovery process is an iterative sequence of the following steps

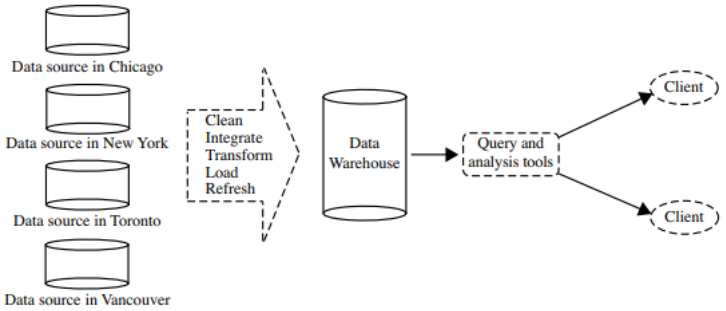
1. Data cleaning (to remove noise and inconsistent data)
2. Data integration (where multiple data sources may be combined)

3. Data selection (where data relevant to the analysis task are retrieved from the database)
4. Data transformation (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)
5. Data mining (an essential process where intelligent methods are applied to extract data patterns)
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on interestingness measures
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present mined knowledge to users)

Association rule mining is a data mining technique that finds patterns and relationships between items in large datasets.



A **data warehouse** is a repository of information collected from multiple sources, stored under a unified schema, and usually residing at a single site. Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.



Types of Data :

Data Type	Description	Examples	Common Operations
Numerical Data	Data that represents quantities or measurements that can be expressed in numerical form.	Age, height, income, temperature, sales figures	Addition, subtraction, multiplication, division, calculation of averages, medians, and standard deviations

Categorical Data	Data that represents categories or classifications that cannot be meaningfully ordered or measured numerically.	Gender, occupation, city, product category, customer type	Frequency counts, mode (most frequent category), encoding (converting categories into numerical values)
Textual Data	Data that represents human language, including text documents, emails, social media posts, and web pages.	Customer reviews, product descriptions, news articles, social media comments, emails	Text cleaning, tokenization, stemming or lemmatization, feature extraction (e.g., TF-IDF), text classification, sentiment analysis
Temporal Data	Data that represents events or measurements that occur over time.	Timestamps, dates, stock prices, sensor readings, customer purchase history	Time series analysis, trend analysis, seasonality analysis, forecasting
Spatial Data	Data that represents locations or positions in space.	Geographic coordinates, latitude and longitude, street addresses, zip codes, maps	Geographic Information Systems (GIS), spatial analysis, location-based services, routing and navigation

Data quality

Data quality is a critical aspect in the field of data mining as the accuracy and reliability of the data directly impact the results and insights obtained from the mining process.

Poor data quality can lead to inaccurate or biased conclusions, hinder the performance of algorithms, and compromise the effectiveness of data-driven decision-making.

Data Quality Dimension	Description	Impact on Data Mining
Accuracy	The degree to which the data reflects the real-world entities or phenomena it represents.	Inaccurate data can lead to biased models and inaccurate predictions.
Completeness	The extent to which the data is free from missing or incomplete values.	Missing values can distort patterns and hinder the performance of data mining algorithms.
Consistency	The absence of contradictions or inconsistencies within the data, and across different data sources.	Inconsistencies can lead to unreliable results and hinder the ability to make informed decisions.
Timeliness	The currency and relevance of the data to the current context.	Outdated data may not reflect current trends and patterns, leading to inaccurate insights.
Uniqueness	The absence of duplicate or redundant data points within the dataset.	Duplicate data can inflate data counts and distort the true distribution of values.
Validity	The conformity of the data to the intended purpose and usage scenarios.	Invalid data can lead to misleading conclusions and hinder the ability to address specific business problems.

▼ *2. Conditional Probability and Bayesian Classification*

Conditional probability is a fundamental concept in probability theory and statistics. It deals with the probability of an event occurring given that another event has already occurred. It allows us to adjust our probability calculations based on new information or conditions. Conditional probability is denoted as $P(A|B)$, where:

- $P(A)$: The probability of event A occurring.
- $P(B)$: The probability of event B occurring.
- $P(A \cap B)$: The probability that both events A and B occur simultaneously, known as the joint probability of A and B.
- $P(A|B)$: The conditional probability of event A occurring given that event B has occurred.
- $P(B|A)$: The conditional probability of event B occurring given that event A has occurred.

Conditional Probability

$$P(A \mid B) = P(A \cap B) / P(B)$$

$$P(B \mid A) = P(A \cap B) / P(A)$$

Bays Theorem

$$P(A \cap B) = P(A \cap B)$$

P(A / B) * P(B) = P(B / A) * P(A)

P(A / B) = P(B / A) * P(A) / P(B)

A : Hypothesis	***B : Data***
P(A) : Prior	***P(B) : Marginal***
P(A / B) : Posterior	***P(B / A) : Likelihood***

To understand conditional probability better, consider a few examples:

1. **Coin Toss Example:**
   - Event A: Getting a heads (H) when tossing a fair coin.
   - Event B: Getting a coin toss result (H or T).
   - P(A) = 1/2 (since there are two equally likely outcomes: H and T).
   - P(B) = 1 (because you will always get either H or T when you toss the coin).
   - P(A ∩ B) = 1/2 (because if you have already tossed the coin, you know the result is either H or T).
   - So, P(A|B) = P(A ∩ B) / P(B) = (1/2) / 1 = 1/2.

   This means that if you already know the result of the coin toss (event B has occurred), the probability of getting a heads (event A) is 1/2.
2. **Card Deck Example:**
   - Event A: Drawing an ace from a standard deck of 52 cards.
   - Event B: Drawing a red card (hearts or diamonds).
   - P(A) = 4/52 (there are four aces in a deck of 52 cards).
   - P(B) = 26/52 (half the deck is red cards).
   - P(A ∩ B) = 2/52 (there are two red aces: one in hearts and one in diamonds).
   - So, P(A|B) = P(A ∩ B) / P(B) = (2/52) / (26/52) = 2/26 = 1/13.

   If you know that you've drawn a red card (event B has occurred), the probability of drawing an ace (event A) is 1/13.

Q2. How the Bayes theorem is useful for classification in data mining? Suppose the fraction of UG students who smoke is 15% and the fraction of PG students who smoke is 23%. If one-fifth of the college students are PG students and the rest are UG students, If a student smokes, is he/she more likely to be a UG or PG student? (Note: Solve using Bayesin classification). **(04)**

Solution :

S = student who are Smoking, UG = UG student, PG = PG student

P(S/UG) = 15/100 = 0.15 (Given)

P(S/PG) = 23/100 = 0.23 (Given)

P(PG) = 1/5 = 0.2 (Given)

P(UG) = Total student - P(PG) = 1 - 0.2 = 0.8

We have to fine P(UG/S), P(PG/S) which is greater

P(UG/S) = P(S/UG) * *P(UG) / P(S) = 0.15 **** 0.8 / P*(S) = 0.12* / P*(S)*

P(PG/S) = P(S/PG) * *P(PG) / P(S) = 0.23 **** 0.2 / P*(S)* = 0.046/ P*(S)*

Ans : P(UG/S) > P(PG/S)

Bayesian Classification

Bayesian classification is a probabilistic approach to machine learning and pattern classification. It is based on the principles of Bayesian probability theory and is used for various classification tasks, such as spam email detection,

document categorization, medical diagnosis, and more. The core idea behind Bayesian classification is to use Bayes' theorem to estimate the probability of a data point belonging to a particular class based on the available evidence or features.

Here's how Bayesian classification works:

1. **Training Phase**:
   - During the training phase, the algorithm learns from a labeled dataset, which consists of examples with known class labels.
   - It calculates the prior probabilities of each class, which represent the likelihood of each class occurring in the dataset.
   - It also estimates the conditional probabilities of each feature given each class. These are often referred to as class-conditional probabilities or likelihoods.
   - In simpler terms, during training, the algorithm learns how often each class occurs and how features are distributed within each class.
2. **Classification Phase**:
   - In the classification phase, the algorithm uses the information learned during training to make predictions or classify new, unseen data points.
   - Given a new data point with a set of features, the algorithm calculates the posterior probabilities of the data point belonging to each class using Bayes' theorem.
   - The class with the highest posterior probability is chosen as the predicted class for the data point.

Mathematically, the formula for Bayesian classification is:

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

Where:

- $P(C_k|x)$: The posterior probability that the data point belongs to class C_k given the observed features x.
- $P(x|C_k)$: The likelihood of observing the features x given class C_k. This is based on the conditional probabilities learned during training.
- $P(C_k)$: The prior probability of class C_k. This represents the probability of class C_k occurring in the training dataset.
- $P(x)$: The probability of observing the features x. This is a normalization constant that ensures that the probabilities sum to 1.

Bayesian classification is particularly useful when dealing with small datasets or situations where you want to incorporate prior knowledge into the classification process. It also provides a probabilistic framework for handling uncertainty, making it robust for tasks like text classification, where word occurrences can be modeled probabilistically.

Common variations of Bayesian classification include Naive Bayes, which assumes that features are conditionally independent given the class (even if that assumption is not entirely accurate in practice, it often works well), and Bayesian networks, which extend the approach to model dependencies between features explicitly.

need a example from class note book

How the Bayes theorem is useful for classification in data mining ?

Bayes' theorem is extremely useful for classification in data mining and machine learning because it provides a probabilistic framework for making decisions based on evidence or data. It allows you to estimate the probability of a data point belonging to a particular class given its features, making it a powerful tool for various classification tasks. Here's how Bayes' theorem is beneficial in data mining for classification:

1. **Probabilistic Modeling:** Bayes' theorem enables you to model the conditional probabilities of different classes given the available features. This modeling allows you to incorporate uncertainty into the classification process. Instead of making binary decisions (e.g., "Is it spam or not?"), you can calculate the probability of each class, providing a more nuanced and probabilistic view of the classification.
2. **Incorporating Prior Knowledge:** You can use prior probabilities (prior beliefs) to influence the classification. For example, if you have prior knowledge about the likelihood of certain events or classes occurring, you can incorporate this information into the classification. This is particularly useful when dealing with imbalanced datasets or when you have domain-specific knowledge.
3. **Handling Uncertainty:** In real-world scenarios, data can be noisy or incomplete. Bayes' theorem can handle this uncertainty gracefully by modeling the probability of observing specific features given each class. It can provide reasonable predictions even when some features are missing or when there is uncertainty in the data.
4. **Feature Selection and Feature Engineering:** Bayesian classification can guide feature selection and feature engineering efforts. By calculating the conditional probabilities of features given each class, you can identify which features are most informative for distinguishing between classes. This can help in feature selection, dimensionality reduction, and improving model performance.
5. **Text Classification:** In natural language processing tasks like text classification (e.g., sentiment analysis, spam detection, document categorization), Bayesian approaches, such as Naive Bayes, are commonly used. They can model the likelihood of observing words or phrases in different classes, making them effective for text-based classification.
6. **Incremental Learning:** Bayes' theorem allows for incremental learning. As new data becomes available, the model can be updated to incorporate the new evidence and adjust the probabilities accordingly. This is useful in scenarios where the data distribution changes over time.
7. **Multiclass Classification:** Bayes' theorem can handle multiclass classification problems naturally. It allows you to estimate the probabilities of each class, and the class with the highest probability can be selected as the predicted class.
8. **Model Interpretability:** Bayesian classification provides insights into the reasons behind a particular classification decision. You can examine the conditional probabilities and see which features influenced the classification outcome the most, making the model more interpretable.
9. **Ensemble Methods:** Bayesian classification can be integrated into ensemble methods like Bayesian Model Averaging (BMA) or Bayesian AdaBoost, which combine multiple Bayesian models to improve classification performance.

Overall, Bayes' theorem and its related techniques are valuable tools in data mining and machine learning because they offer a principled way to handle uncertainty, incorporate prior knowledge, and make probabilistic decisions, which can be particularly useful in complex and uncertain real-world scenarios.

▼ *3. Unsupervised Learning > Association Rule Mining*

Association rule mining is a data mining technique used in machine learning and data analysis to discover interesting patterns and relationships within datasets. It specifically focuses on finding associations or relationships between items in large transactional databases, where items could be products in a store, web pages visited by a user, or any other entities that can be grouped into transactions.

The most common application of association rule mining is in market basket analysis, where the goal is to discover which items are frequently purchased together by customers. For example, a typical result of association rule mining might reveal that customers who buy bread and milk are also likely to buy eggs.

Here are some key concepts and terms related to association rule mining:

1. **Itemset :** An itemset is a collection of one or more items that appear together in a transaction. For example, {milk, bread} is an itemset representing the purchase of both milk and bread.

   Item Set = { I1, I2, I3, I4, . . . , Ik }

   Total number of Item set possible with ‘k’ number of item = 2^k-1

2. **Support Count :** The frequency of Item occurrence of an item set

   SC(x) = Freq(x), x = particular item or items from itemset.

   Minimum support count : it refers to the minimum number of occurrences or transactions in which an itemset or association rule must be present to be considered as frequent and included in the analysis. This parameter helps filter out less significant patterns and focuses on those with a sufficient level of support in the dataset.

   Let t = threshold or minimum support count

   If SC(x) ≥ t, then included 'x' in the analysis

   Frequent Item Set : The items which are satisfied minimum support count.

3. **Support :** The support of an itemset is the proportion of transactions in which the itemset appears. It indicates how frequently an itemset occurs in the dataset.

   S**(x) = (Freq(x) or** SC(x)) **/ T**

   Freq(x) = frequency of items occur in the transaction

   T = Number of transaction

4. **Confidence :** The confidence of an association rule measures the likelihood (সম্ভাবনা) that an item (or itemset) B is purchased when item (or itemset) A is purchased. It is defined as the support of both A and B divided by the support of A.

   Conf(X⇒Y) = SC(x , y) / SC(x) or, **S(x , y) / S(x)**

   Confidence Threshold or Minimum Confidence : It is a parameter used in association rule mining to filter and select association rules that meet a certain level of confidence

   - If you set a high confidence threshold (e.g., 0.8 or 80%), it means you want to select only those association rules where the likelihood of item B being purchased when item A is purchased is very high. This results in a more stringent selection of rules, typically leading to fewer but more reliable rules.
   - If you set a lower confidence threshold (e.g., 0.5 or 50%), you'll include association rules with a lower level of confidence. This can result in a larger number of rules but may include weaker associations.

5. **Rules** : Here **X⇒Y mean X determine Y and { X, Y} ∈ itemset and X ∩ Y = null**

   Total number of Rules possible with 'k' number of item = 3^k+1-2^(k+1)

6. **Lift :** Lift is a measure of how much more likely item B is purchased when item A is purchased compared to when item B is purchased independently of item A. A lift value greater than 1 indicates a positive association, while a lift value less than 1 indicates a negative association.

   Lift(X⇒Y) = S(x ,y) / (S(x) * S(y)) or, Conf(X⇒Y) / S(y)

The association rule mining process typically involves the following steps:

1. **Data Collection:** Gather transactional data that contains information about items purchased together.
2. **Data Preprocessing:** Clean and preprocess the data, including removing duplicates and handling missing values.
3. **Frequent Itemset Generation:** Identify item sets that meet a minimum support threshold. These are considered frequent item sets.
4. **Association Rule Generation:** Create association rules from the frequent item sets by calculating confidence and other relevant metrics.
5. **Rule Evaluation:** Evaluate the generated rules using metrics like confidence, lift, and support. Select the most interesting and relevant rules based on these metrics.
6. **Visualization and Interpretation:** Visualize and interpret the discovered association rules to gain insights into the data.
7. **Application:** Apply the discovered rules in real-world scenarios, such as optimizing product placement in a store or making personalized product recommendations to customers.

Association rule mining algorithms include **Apriori, FP-growth (Frequent Pattern-growth), and Eclat, among others.** These algorithms help automate the process of finding meaningful associations in large datasets and have applications in various domains, including retail, marketing, and recommendation systems.

Example

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

1. **Itemset :** { Bread, Diaper, Beer, Eggs, Milk, Coke }

   Total number of Item set possible with '6' number of item = 2^6-1 = 63

2. **Support Count :** SC(x) = Freq(x)

   SC(Bread) = 4

   SC(Diaper) = 4

   SC(Beer) = 3

   SC(Eggs) = 1

   SC(Milk) = 4

   SC(Coke) = 2

   SC(Bread, Diaper, Coke) = 1

3. **Support :** S**(x) = (Freq(x) or** SC(x)) **/ T**

   S(Bread, Diaper, Coke) = SC(Bread, Diaper, Coke) / Number of Transaction

   = 1 / 5

   = 0.2

4. **Confidence : Conf(X⇒Y) = SC(x , y) / SC(x)** or, **S(x , y) / S(x)**

   Conf({ Bread, Diaper } ⇒ {Coke})

   = SC{ Bread, Diaper, Coke } **/ SC(** Coke **)**

   = 1 / 2

   = 0.5 or 50%

5. **Lift : Lift(X⇒Y)** = S(x ,y) / (S(x) * S(y)) or, Conf(X⇒Y) / S(y)

   Lift({ Bread, Diaper } ⇒ {Coke})

   = S { Bread, Diaper, Coke } **/ (S{** Bread, Diaper } *** S{** Coke **})**

   = 0 . 2 / (3/ 5 ∗ 1/ 5)

   = **0.12**

nikhil pdf page 7 to 9 example

▼ *4. **Association Rule Mining> Apriori** Algorithm*

nikhil pdf page 9 to 11 example

https://www.youtube.com/watch?v=g9VASv1Q-_I

minimum support count is 2

minimum confidence is 60%

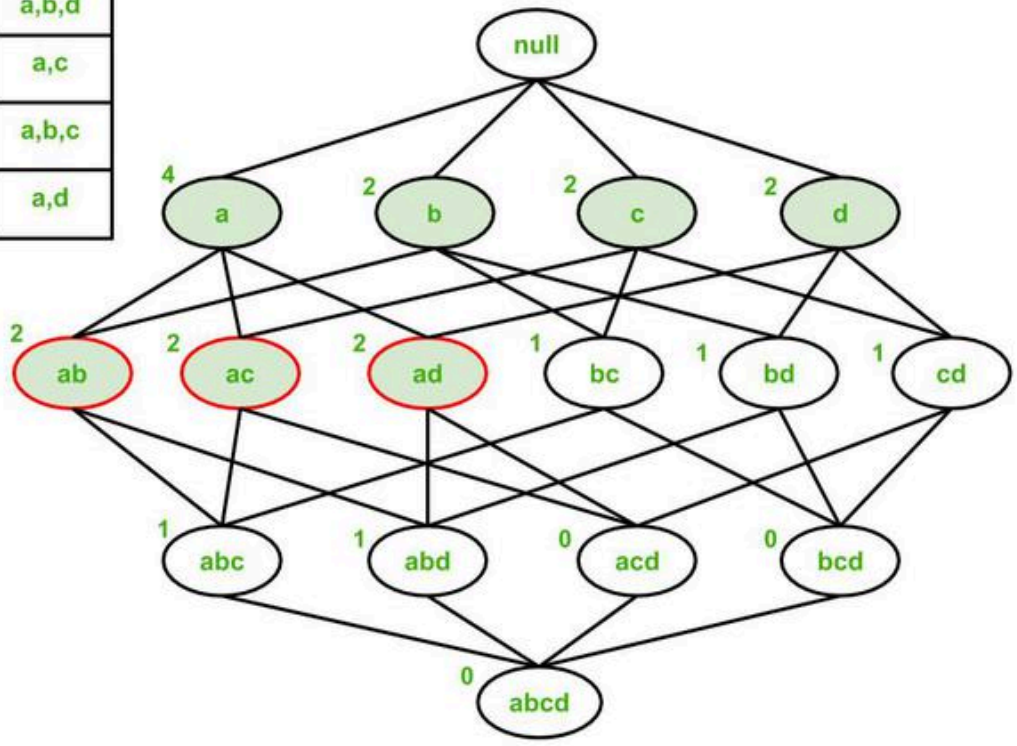
TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Steps

1. Create a table containing support count of each item present in dataset – Called **C1(candidate set)**

Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

TID	Items
1	a,b,d
2	a,c
3	a,b,c
4	a,d



Maximal Frequent Itemset

A **maximal frequent itemset** is a frequent itemset for which none of its immediate supersets are frequent.

Closed Frequent Itemset

A frequent itemset is closed, when no (immediate) superset has the same support.

▼ 5. *Association Rule Mining > FP-growth (Frequent Pattern-growth)*

https://www.geeksforgeeks.org/ml-frequent-pattern-growth-algorithm/

Conditional Pattern Base (CPB) :

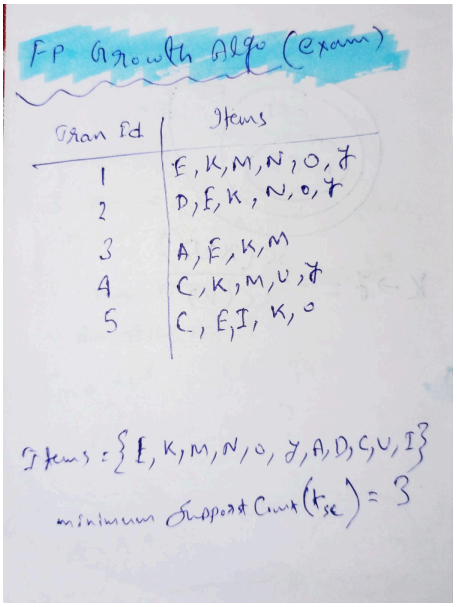
The Conditional Pattern Base is computed by collecting the path labels of all paths that lead to any node of the given item in the frequent-pattern tree

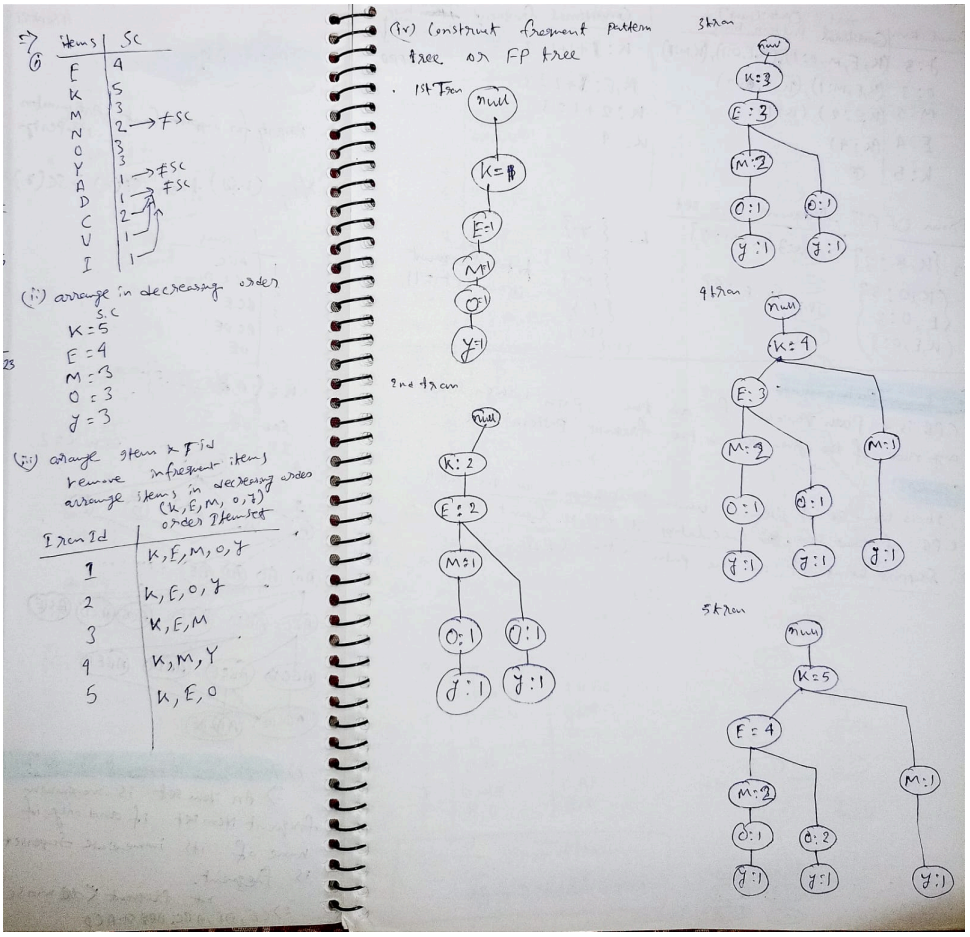
Conditional Frequent Pattern Base (CFPB) :

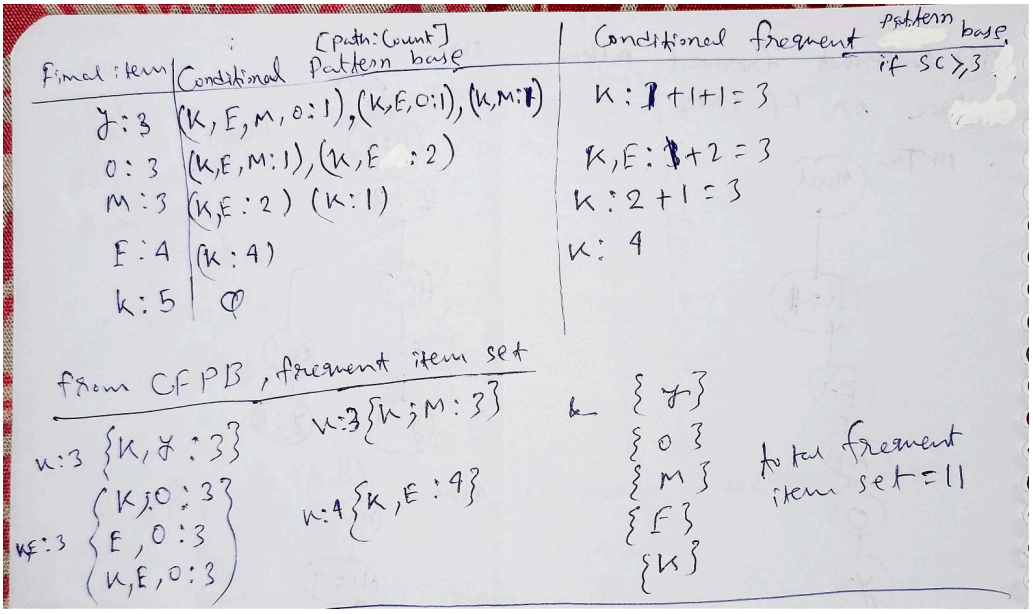
It is done by taking the set of elements that is common in all the paths in the Conditional Pattern Base of that item and calculating its support count by summing the support counts of all the paths in the Conditional Pattern Base.

Maximum number of rules = 3^k - 2^(k+1) +1

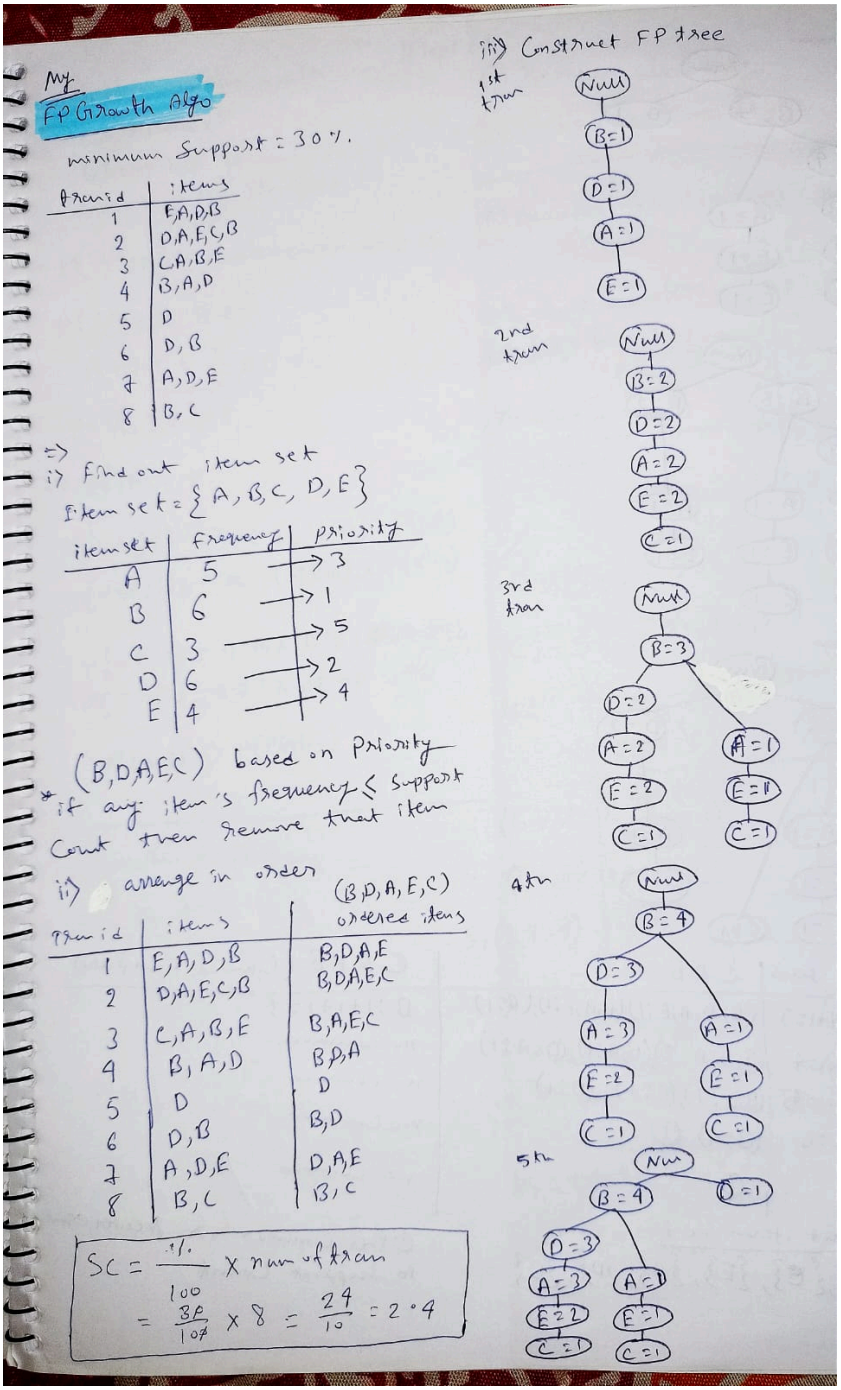
Example 1 (class example)

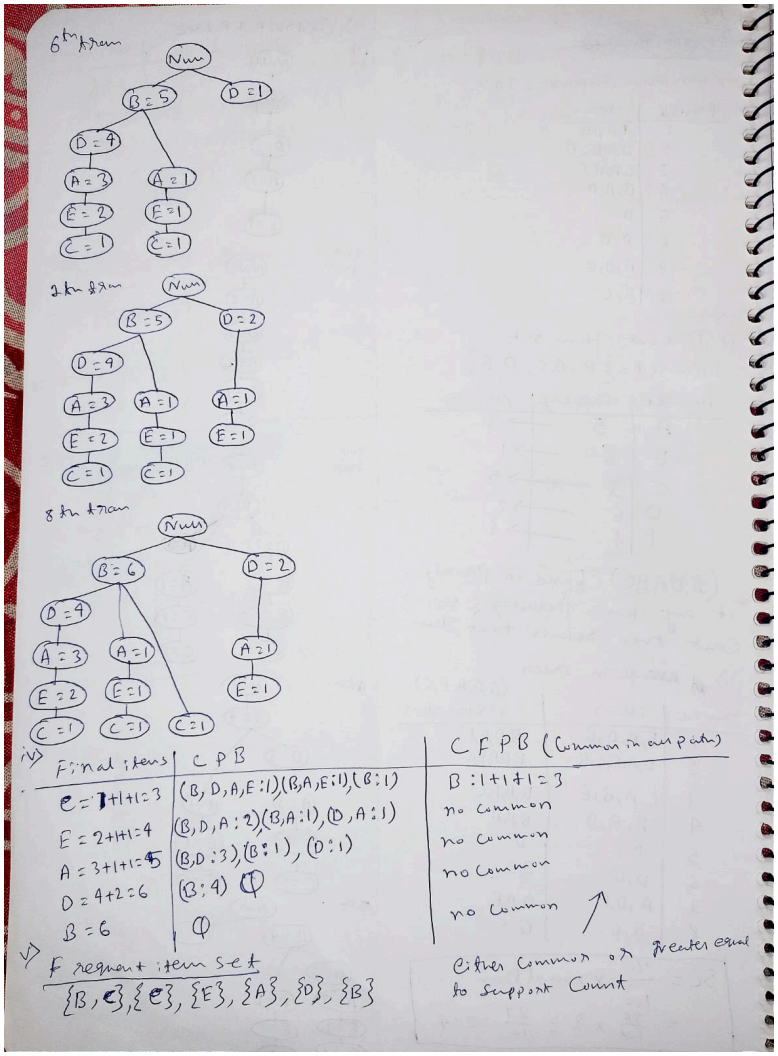






Example 2 (youtube example)


My
FP Growth Algo
minimum Support = 30%.
Trans id | items
1 E,A,D,B
2 D,A,E,C,B
3 C,A,B,E
4 B,A,D
5 D
6 D,B
7 A,D,E
8 B,C
=>
i) Find out item set
Item set = {A, B, C, D, E}
item set | Frequency | priority
A 5 → 3
B 6 → 1
C 3 → 5
D 6 → 2
E 4 → 4
(B,D,A,E,C) based on priority
* if any item's frequency ≤ support Count then remove that item
ii) arrange in order
Trans id | items | (B,D,A,E,C) ordered items
1 E,A,D,B B,D,A,E
2 D,A,E,C,B B,D,A,E,C
3 C,A,B,E B,A,E,C
4 B,A,D B,D,A
5 D D
6 D,B B,D
7 A,D,E D,A,E
8 B,C B,C
SC = %/100 × num of trans
= 30/100 × 8 = 24/10 = 2.4
iii) Construct FP tree
1st tran
Null
B=1
D=1
A=1
E=1
2nd tran
Null
B=2
D=2
A=2
E=2
C=1
3rd tran
Null
B=3
D=2
A=2
E=2
C=1
A=1
E=1
C=1
4th
Null
B=4
D=3
A=3
E=2
C=1
A=1
E=1
C=1
5th
Null
B=4
D=1
D=3
A=3
E=2
C=1
A=1
E=1
C=1



▼ *6. Unsupervised Learning > Clustering*

Clustering is a fundamental technique in machine learning and data analysis that involves grouping similar data points or objects together based on their inherent characteristics or features. The primary goal of clustering is to discover meaningful patterns or structures within a dataset without prior knowledge of the class labels or categories. Clustering is an unsupervised learning technique, as it does not rely on labeled data for training.

Here are the key concepts and aspects of clustering in machine learning:

1. **Objective**:
   - The primary objective of clustering is to partition a dataset into clusters or groups in such a way that data points within the same cluster are more similar to each other than to those in other clusters.
   - It aims to uncover hidden structures or relationships in the data, which can be valuable for data exploration and analysis.
2. **Similarity Measure**:
   - Clustering algorithms use a similarity or distance measure to quantify the similarity between data points. Common distance metrics include Euclidean distance, cosine similarity, and Jaccard similarity, depending on the nature of the data.
3. **Types of Clustering Algorithms**:
   - There are various clustering algorithms, each with its own characteristics and applications. Some popular clustering algorithms include:

- **K-Means:** Divides the data into a specified number (k) of clusters, minimizing the within-cluster sum of squares.
- **Hierarchical Clustering:** Forms a hierarchical tree-like structure of clusters, known as a dendrogram, where the number of clusters can be adjusted.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Identifies clusters based on data density, allowing for the discovery of irregularly shaped clusters and noise points.
- **Agglomerative Clustering:** A hierarchical clustering approach that starts with individual data points and gradually merges clusters.
- **Mean Shift:** A mode-seeking clustering algorithm that identifies clusters by locating high-density regions in the data.
- **Gaussian Mixture Models (GMM):** Models data points as a mixture of Gaussian distributions and estimates cluster parameters using the Expectation-Maximization (EM) algorithm.
- **Spectral Clustering:** Uses the spectral decomposition of an affinity matrix to partition data into clusters.
- **Self-Organizing Maps (SOM):** A neural network-based approach that creates a low-dimensional representation of the data, often used for visualizing high-dimensional data.

4. **Choosing the Number of Clusters (k)**:
   - In many clustering algorithms, you need to specify the number of clusters (k) in advance. Selecting an appropriate k value is often a critical step and may require techniques like the elbow method, silhouette analysis, or domain knowledge.
5. **Evaluation**:
   - Clustering results can be evaluated using various metrics, although the lack of ground truth labels makes evaluation challenging. Common evaluation metrics include silhouette score, Davies-Bouldin index, and Dunn index.
6. **Applications**:
   - Clustering is used in a wide range of applications, including customer segmentation, document clustering, image segmentation, recommendation systems, anomaly detection, and more.
7. **Limitations**:
   - Clustering can be sensitive to the choice of distance metric, the initial configuration of centroids (in k-means), and the inherent structure of the data.
   - It may not always produce meaningful results if the data does not exhibit clear cluster boundaries.

Clustering is a powerful tool for data exploration and analysis, helping to uncover patterns and structures in complex datasets. The choice of clustering algorithm and parameter settings depends on the specific characteristics of the data and the goals of the analysis.

Distance measures or similarity measures

Euclidean Distance =

Formula: $d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Manhattan Distance (City Block Distance) =

Formula: $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$

L∞ (L-infinity) distance =

$$d_\infty(P, Q) = \max_{i=1}^{n} |p_i - q_i|$$

Ex x = {1, 5, 7}, y = {2, 8, 6}

Sample	f1	f2	f3
x	1	5	7
y	2	8	6

Euclidean Distance(x, y) = √[(1-2)^2+(5-8)^2+(7-6)^2] = √(1+9+1) = √11 = 3.31

Manhattan Distance (City Block Distance) = 1+3+1 = 5

L∞ (L-infinity) distance(x, y) = max(1, 3, 1) = 3

▼ *7. Clustering > K-Means, Error Calculation*

Algorithm

1. Specify the number of clustering that is k
2. Randomly choose initially K-Centroid
3. Repeat 4, 5
4. Assign each point to it's closest centroid or Mean
5. Compute new centroid of each Cluster
6. Stop if centroid are not changing
7. Calculate Error (usually error will decrease in every iteration)

Time Complexity= O (n * m * I * K)

I = Number of iteration

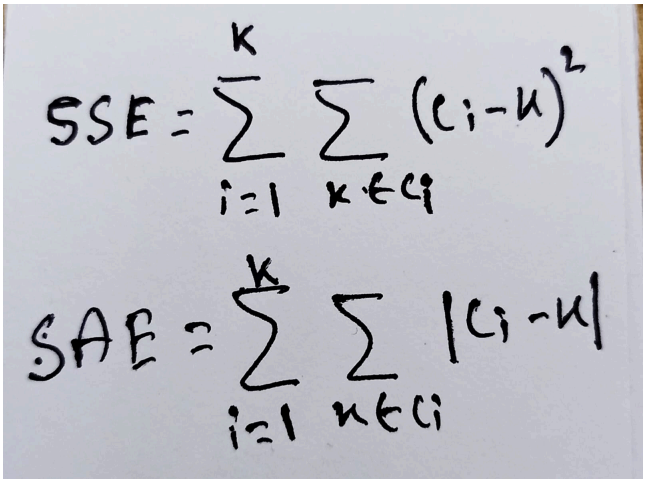
K = Number of Cluster

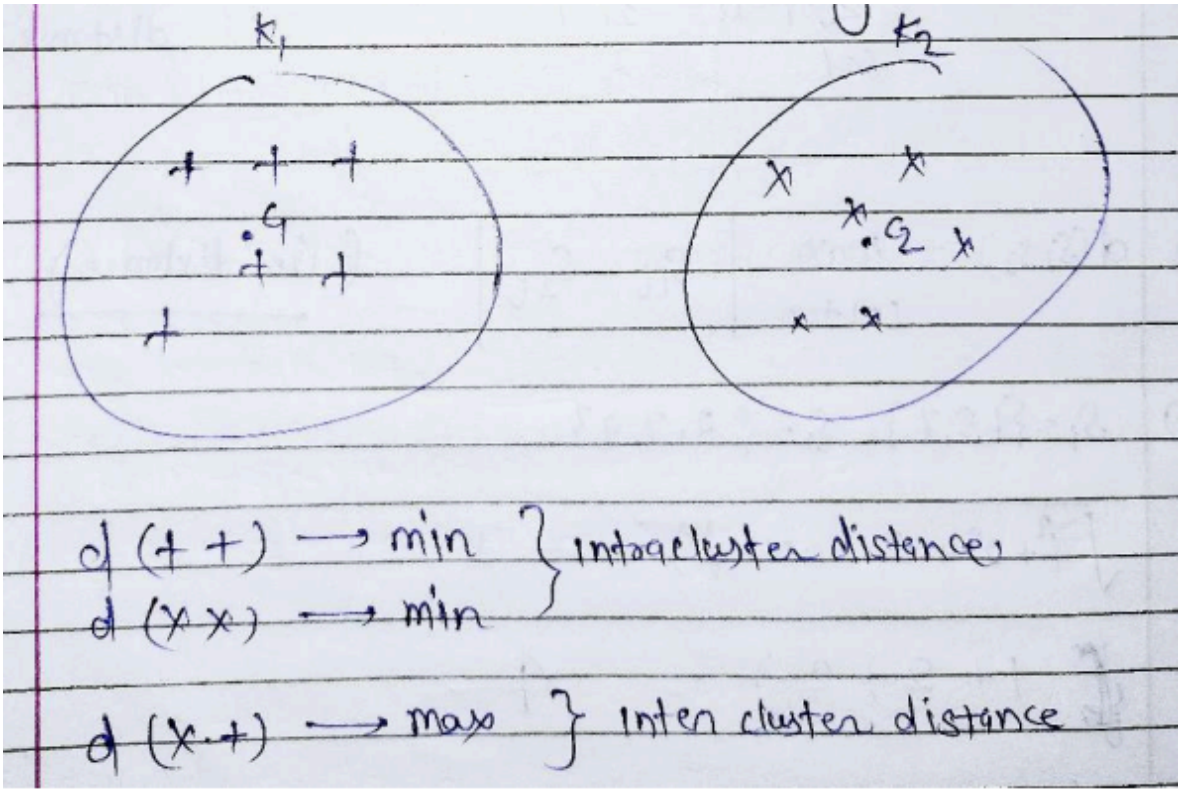
n = Number of Sample

m = Number of Feature

Error Calculation Method (Apply any one)

1. Sum of Squared Error (SSE)
2. Sum of Absolute error (SAE)





Example

Given

Sample	F1	F2	F3
S1	1	2	3
S2	3	5	8
S3	2	4	7
S4	6	1	8
S5	20	46	60
S6	18	32	40

Number of Cluster K = 2

and S3 and S5 are our initial cluster k1, k2.

using **Manhattan Distance**

Solution

1st Round

Sample	k1=S3	k2=S5	Comment
S1	1+2+4=7	19+44+57=120	closest S3, so includes in k1
S2	1+1+1=3	17+41+53=110	closest S3, so includes in k1
S3	0	no need to calculate as ‘0’ is minimum	
S4	4+3+1=8	14+45+52=111	closest S3, so includes in k1
S5	no need to calculate as ‘0’ is minimum	0	
S6	16+28+33=77	2+14+20=36	closest S5, so includes in k2

New k1 = { S3, S1, S2, S4 } , k2= { S5, S6 }

Find new centroid or Mean

Mean(k1) = [(1+3+2+6)/4 , (2+5+4+1)/4 , (3+8+7+8)/4]= (3, 3, 6.5)

Mean(k2) = ((20+18)/2, (46+32)/2, (60+40)/2) = (19, 39, 50)

Error Calculation apply SSE

= [(3 - 1)^2+(3 - 2)^2+ (6.5 - 3)^2]

+ [(3 - 3)^2+(3 - 5)^2+ (6.5 - 8)^2]

+ [(3 - 2)^2+(3 - 4)^2+ (6.5 - 7)^2]

+ [(3 - 6)^2+(3 - 1)^2+ (6.5 - 8)^2]

+ [(19 - 20)^2+(39 - 46)^2+(50 - 60)^2]

+ [(19 - 18)^2+(39 - 32)^2+(50 - 40)^2]

= 341

- Error will be decreasing with every new centroid

2nd Round

Sample	k1 = (3, 3, 6.5)	k2 = (19, 39, 50)	Comment
S1 (1,2,3)	|1−3|+ |2−3|+ |3−6.5| = 2 + 1 + 3.5 = 6.5	|1−19|+ |2−39|+ |3−50| = 18 + 37 + 47 = 102	closest to k1
S2 (3,5,8)	|3−3|+ |5−3|+ |8−6.5| = 0 + 2 + 1.5 = 3.5	|3−19|+ |5−39|+ |8−50| = 16 + 34 + 42 = 92	closest to k1
S3 (2,4,7)	|2−3|+ |4−3|+ |7−6.5| = 1 + 1 + 0.5 = 2.5	|2−19|+ |4−39|+ |7−50| = 17 + 35 + 43 = 95	closest to k1
S4 (6,1,8)	|6−3|+ |1−3|+ |8−6.5| = 3 + 2 + 1.5 = 6.5	|6−19|+ |1−39|+ |8−50| = 13 + 38 + 42 = 93	closest to k1
S5 (20,46,60)	|20−3|+ |46−3|+ |60−6.5| = 17 + 43 + 53.5 = 113.5	|20−19|+ |46−39|+ |60−50| = 1 + 7 + 10 = 18	closest to k2
S6 (18,32,40)	|18−3|+ |32−3|+ |40−6.5| = 15 + 29 + 33.5 = 77.5	|18−19|+ |32−39|+ |40−50| = 1 + 7 + 10 = 18	closest to k2

Final clusters remain:

- **k1** = { S3, S1, S2, S4 }
- **k2** = { S5, S6 }

Find new centroid or Mean

Mean(k1) = [(1+3+2+6)/4 , (2+5+4+1)/4 , (3+8+7+8)/4]= (3, 3, 6.5)

Mean(k2) = ((20+18)/2, (46+32)/2, (60+40)/2) = (19, 39, 50)

No change means **the algorithm has converged**.

Error Calculation apply SSE

= 341

▼ *8. Clustering > Nearest Neighbor Clustering*

Sample	F1	F2
S1	2	10
S2	2	5
S3	8	4
S4	5	8

Use Euclidean Distance

Threshold = 4

S5	7	5
S6	6	4
S7	1	2
S8	4	9

Let Cluster k1 = S1 = (2,10)

Check S2 belongs to k1 cluster or not

d(k1 , S2) = ((2-2)^2+(10-5)^2)^(1/2) = 5 > T**hreshold, not satisfy**

Create different Cluster with S2(S2 does not belong to k1),

k2 = S2 = (2,5)

now we have 2 cluster k1, k2

Check S3 belongs to k1 or K2 cluster or not

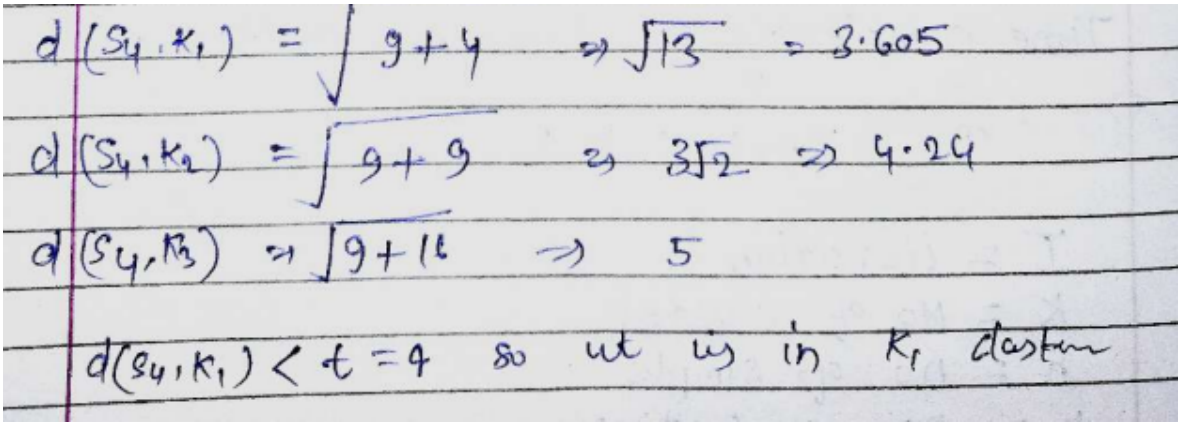
d(k1 , S3)= ((2-8)^2+(10-4)^2)^(1/2) = 8.48 > **Threshold, not satisfy**

d(k2 , S3)= ((2-8)^2+(5-4)^2)^(1/2) = 6 > **Threshold, not satisfy**

Create different Cluster with S3,

k3 = S3 = (8,4)

Check S4 belongs to (k1 or k2 or k3) cluster or not



k1 = { S1, S4 }

k2 = { S2 }

k3 = { S3 }

Check S4 belongs to (k1 or k2 or k3) cluster or not

d(S5, k1)=min[d(S5 , S1) , d(S5 , S4)] = ?

d(S5, k2) = d(S5 , S2) = ?

d(S5, k3) = d(S5 , S3) = ?

similarly calculate s5, s6, s7, s8

- if any point belongs to 2 different cluster then take minimum one

▼ *9. Clustering > Agglomerative/Hierarchical Clustering*

page 23 to 30 pending

Algorithm

1. Compute Distance matrix.
2. Repeat 3, 4.
3. Marge the closest (minimum value from matrix) two cluster.
4. Update the distance matrix to reflect the distance between to new cluster and original cluster.

5. Do the above steps, until only one cluster remaining.
6. Create Dendrogram

Types of link or Marge

1. Single link or Min link

   d [s1, (s2, s3)] = Min [d(s1, s2), d(s1, s3)]

2. Complete Link or Max Link

   d [s1, (s2, s3)] = Max [d(s1, s2), d(s1, s3)]

3. Centroid

   d(s1,(s2,s3))

   let centroid of (s2,s3) = [(x1+y1)/2 , (x2+y2)/2]

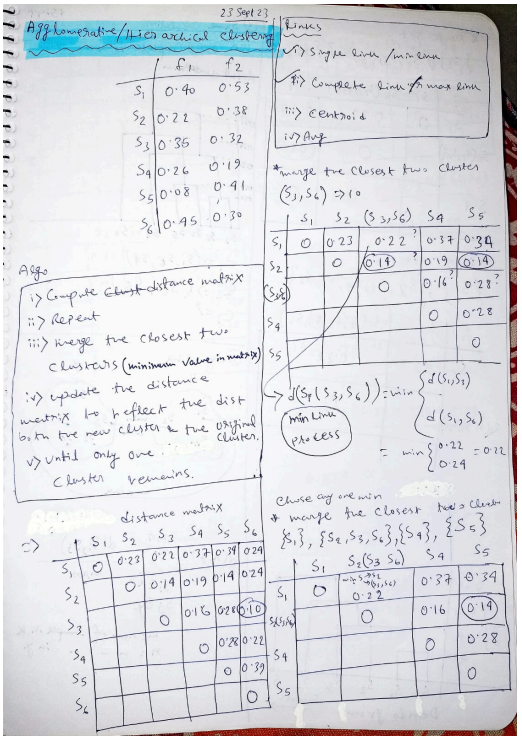
   d(s1,(s2,s3)) = d(s1, [(x1+y1)/2 , (x2+y2)/2])

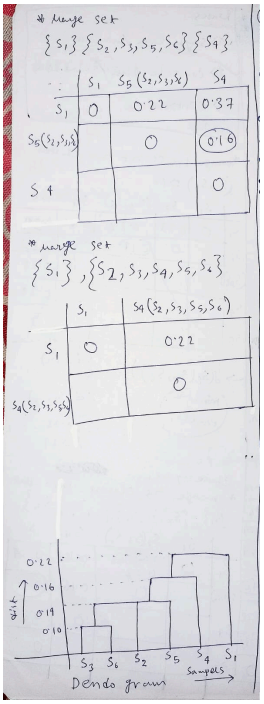
4. Average Link

   d [(s1, s2), (s3 , s4, s5)]

   = [d(s1, s3) + d(s1, s4) + d(s1, s5) + d(s2, s3) + d(s2, s4) + d(s2, s5)] / (2 * 3)

EX : Single Link





Same for Other Link

Advantage and Disadvantage

	Strength	***Limitations***
Single link or Min link	Can handle non-elliptical shapes	Sensitive to noise and outliers
MAX or Complete Link	Less susceptible to noise and outliers	1. Tends to break large clusters 2. Tends to break large clusters
Group Average or Average Link	Less susceptible to noise and outliers	Biased towards globular clusters
Centroid Distance or Ward's Method	Less susceptible to noise and outliers	Biased towards globular clusters

Time and Space Requirements

- ***O(N2) space since it uses the distance matrix.***
  N is the number of points.
- ***O(N3) time in many cases***

There are N steps in the clustering process. At each step, the size of the N^2 distance matrix must be updated and searched.

- However, some approaches can reduce the time complexity to O(N^2 * log(N)), making it more efficient for certain scenarios.

Overall Strengths of Hierarchical Clustering

1. Hierarchical clustering does not require assuming a specific number of clusters in advance, making it more flexible.
2. You can obtain any desired number of clusters by "cutting" the dendrogram at the appropriate level, allowing for adaptability in cluster count.
3. Hierarchical clusters can correspond to meaningful taxonomies or categories, such as in the context of online shopping websites. For example, categories like "electronics" (containing subcategories like "computer" and "camera"), "furniture," and "groceries" can represent meaningful groupings or categories within the context of online shopping.

Problems and Limitations

1. Once two clusters are combined in hierarchical clustering, it is a irreversible process and cannot be undone.
2. Hierarchical clustering does not have a direct objective function to minimize during the clustering process.

3. Different linkage schemes in hierarchical clustering have issues related to sensitivity to noise and outliers, difficulty in handling clusters of different sizes and irregular shapes, and the potential for breaking large clusters apart.

▼ *10. **Clustering > DBSCAN(Density-Based Spatial Clustering of Applications with Noise)***

r = Radius (given in the Qn)

Min-Point (given in the Qn)

Different Types of Point

1. Core Point : Count of min point ≥ r
2. Border Point : Count of min point < r and neighbourhood of core point(minimum one core point Included in the set)
3. Noise Point : if not included in Core Point or Border Point
4. Directly Density Reachable :

   A sample s2 is DDR from sample s1

   if s1 is the core point and s2 is in s1 neighbourhood

   s2 → s1 but not s1 → s2

   s2 is DDR from s1, but s1 is not DDR to s2

Algorithm

for each sample S.i ∈ **D**

do,

if, S is not yet classified

then,

If, S is a core point

then, Collect all samples density reachable from S and assign them to a new cluster

Else, we have to assign to noise point

Example

Given Data

r = 0.6, min point = 4

Use **Euclidean Distance**

Sample	F1	F2
S1	1	2
S2	3	4
S3	2.5	4
S4	1.5	2.5
S5	3	5
S6	2.8	4.5
S7	2.5	4.5
S8	1.2	2.5
S9	1	3
S10	1	5
S11	1	2.5
S12	5	6
S13	4	3

⇒ Calculate Distance Matrix

Sample	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
S1	[0.0,	2.8,	2.5,	0.7,	3.6,	3.1,	2.9,	0.5,	1.0,	3.0,	0.5,	5.7,	3.2]
S2	[2.8,	0.0,	0.5,	2.1,	1.0,	0.5,	0.7,	2.3,	2.2,	2.2,	2.5,	2.8,	1.4]
S3	[2.5,	0.5,	0.0,	1.8,	1.1,	0.6,	0.5,	2.0,	1.8,	1.8,	2.1,	3.2,	1.8]
S4	[0.7,	2.1,	1.8,	0.0,	2.9,	2.4,	2.2,	0.3,	0.7,	2.5,	0.5,	4.9,	2.5]
S5	[3.6,	1.0,	1.1,	2.9,	0.0,	0.5,	0.7,	3.1,	2.8,	2.0,	3.2,	2.2,	2.2]
S6	[3.1,	0.5,	0.6,	2.4,	0.5,	0.0,	0.3,	2.6,	2.3,	1.9,	2.7,	2.7,	1.9]
S7	[2.9,	0.7,	0.5,	2.2,	0.7,	0.3,	0.0,	2.4,	2.1,	1.6,	2.5,	2.9,	2.1]
S8	[0.5,	2.3,	2.0,	0.3,	3.1,	2.6,	2.4,	0.0,	0.5,	2.5,	0.2,	5.2,	2.8]
S9	[1.0,	2.2,	1.8,	0.7,	2.8,	2.3,	2.1,	0.5,	0.0,	2.0,	0.5,	5.0,	3.0]
S10	[3.0,	2.2,	1.8,	2.5,	2.0,	1.9,	1.6,	2.5,	2.0,	0.0,	2.5,	4.1,	3.6]
S11	[0.5,	2.5,	2.1,	0.5,	3.2,	2.7,	2.5,	0.2,	0.5,	2.5,	0.0,	5.3,	3.0]
S12	[5.7,	2.8,	3.2,	4.9,	2.2,	2.7,	2.9,	5.2,	5.0,	4.1,	5.3,	0.0,	3.2]
S13	[3.2,	1.4,	1.8,	2.5,	2.2,	1.9,	2.1,	2.8,	3.0,	3.6,	3.0,	3.2,	0.0]

Apply r = 0.6, check d(si,sj) ≤ r

Sample	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
S1	[0.0,	2.8,	2.5,	0.7,	3.6,	3.1,	2.9,	0.5,	1.0,	3.0,	0.5,	5.7,	3.2]
S2	[2.8,	0.0,	0.5,	2.1,	1.0,	0.5,	0.7,	2.3,	2.2,	2.2,	2.5,	2.8,	1.4]
S3	[2.5,	0.5,	0.0,	1.8,	1.1,	0.6,	0.5,	2.0,	1.8,	1.8,	2.1,	3.2,	1.8]
S4	[0.7,	2.1,	1.8,	0.0,	2.9,	2.4,	2.2,	0.3,	0.7,	2.5,	0.5,	4.9,	2.5]
S5	[3.6,	1.0,	1.1,	2.9,	0.0,	0.5,	0.7,	3.1,	2.8,	2.0,	3.2,	2.2,	2.2]
S6	[3.1,	0.5,	0.6,	2.4,	0.5,	0.0,	0.3,	2.6,	2.3,	1.9,	2.7,	2.7,	1.9]
S7	[2.9,	0.7,	0.5,	2.2,	0.7,	0.3,	0.0,	2.4,	2.1,	1.6,	2.5,	2.9,	2.1]
S8	[0.5,	2.3,	2.0,	0.3,	3.1,	2.6,	2.4,	0.0,	0.5,	2.5,	0.2,	5.2,	2.8]
S9	[1.0,	2.2,	1.8,	0.7,	2.8,	2.3,	2.1,	0.5,	0.0,	2.0,	0.5,	5.0,	3.0]
S10	[3.0,	2.2,	1.8,	2.5,	2.0,	1.9,	1.6,	2.5,	2.0,	0.0,	2.5,	4.1,	3.6]
S11	[0.5,	2.5,	2.1,	0.5,	3.2,	2.7,	2.5,	0.2,	0.5,	2.5,	0.0,	5.3,	3.0]
S12	[5.7,	2.8,	3.2,	4.9,	2.2,	2.7,	2.9,	5.2,	5.0,	4.1,	5.3,	0.0,	3.2]
S13	[3.2,	1.4,	1.8,	2.5,	2.2,	1.9,	2.1,	2.8,	3.0,	3.6,	3.0,	3.2,	0.0]

Apply min point = 4;

Count(Sample) + count(Neighbourhood) ≥ min point

- 1st assign Core then Border

Sample	Neighbourhood	Status	Comment
S1	S8, S11	Border	core point present
S2	S3, S6	Border	core point present
S3	S2, S6, S7	CORE	count(S3, S2, S6, S7) : 4 ≥ 4
S4	S8, S11	Border	core point present
S5	S6	Border	core point present
S6	S2, S3, S5, S7	CORE	
S7	S3, S6	Border	core point present
S8	S1, S4, S9, S11	CORE	
S9	S8, S11	Border	core point present
S10	NULL	Noise	
S11	S1, S4, S8, S9	CORE	

Sample	Neighbourhood	Status	Comment
S12	NULL	Noise	
S13	NULL	Noise	

Let Chose 1st Core point S3 as a Cluster

C1={ S3, S2, S6, S7 }

Remaining points are

{S1, S4, S5, S8, S9, S11}

Out of remaining points S5 neighbourhood under C1

C1 ={ S3, S2, S6, S7 , S5}

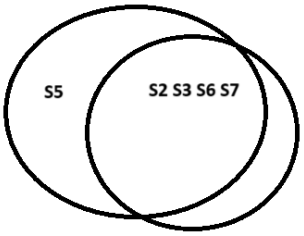
Let Chose 2nd Core point S6 as a Cluster, already included in C1

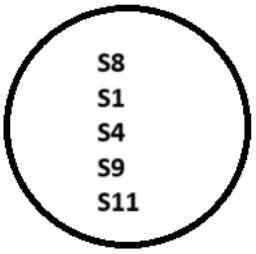
Let Chose 3rd Core point S8 as a Cluster

C2 = { S8, S1, S4, S9, S11 }

Remove Noise Point { S10, S12, S13}

So there are only 2 cluster





C1 ={ S3, S2, S6, S7 , S5 }

C2 = { S8, S1, S4, S9, S11 }

Time Complexity ∝ 1/ Space Complexity

▼ *11. Silhouette Coefficient (SC)*

The Silhouette Coefficient is a measure used to evaluate the quality of clusters created in unsupervised machine learning, such as clustering algorithms like K-Means, Hierarchical Clustering, and DBSCAN. It provides a way to assess how well-separated the clusters are and, therefore, the appropriateness of the chosen number of clusters. The Silhouette Coefficient typically ranges from -1 to +1, with higher values indicating better cluster separation:

- **+1**: A high Silhouette Coefficient indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. This is a sign of a good, well-separated clustering.
- **0**: A score of 0 means that the object is on or very close to the decision boundary between two neighboring clusters, indicating overlapping clusters.
- **1**: A negative Silhouette Coefficient indicates that the object is incorrectly clustered; it may be more similar to a neighboring cluster than to its own. This suggests that the clustering is inappropriate.

Collision (minimize the collision) : inter cluster

Separation (maximize the distance, try to Separate the cluster as much as possible) : outer cluster

SC = (Separation - Collision) / Max(Separation, Collision)

Range of SC always in -1 to 1

Example

k1	A1=(2,5)	A2=(3,4)	A3=(4,6)
k2	B1=(8,3)	B2=(9,2)	B3=(10,5)
k3	C1=(6,10)	C2=(7,8)	C3=(8,9)

Use Manhattan distance

Calculate Distance Matrix

	A1=(2,5)	A2=(3,4)	A3=(4,6)	B1=(8,3)	B2=(9,2)	B3=(10,5)	C1=(

A1=(2,5)	0	1+1=2	2+1=3	6+2=8	7+3=10	8+0=8	4+5=
A2=(3,4)		0	1+2=3	5+1=6	6+2=8	7+1=8	3+6=
A3=(4,6)			0	4+3=7	5+4=9	6+1=7	2+4=
B1=(8,3)				0	1+1=2	2+2=4	2+7=
B2=(9,2)					0	1+3=4	3+8=
B3=(10,5)						0	4+5=
C1=(6,10)							0
C2=(7,8)							
C3=(8,9)							

Calculation for SC(A1)

step 1. Collision of A1 (Inter cluster)

Collision(A1)=[d(A1,A2)+d(A1,A3)] / 2 = (2 + 3)/2=2.5

step 2. Separation of A1 (outer Cluster)

d(A1,k2) = [d(A1,B1)+d(A1,B2)+d(A1,B3)] / 3 = (8+10+8)/3=26/3=8.67

d(A1,k3) = [d(A1,C1)+d(A1,C2)+d(A1,C3)] / 3 = (9+8+10)/3=27/3=9

Separation(A)=MIN [d(A1,k2) ,d(A1,k3)]=8.67

step 3. SC(A1)=(Separation - Collision) / Max(Separation, Collision)

= (8.67-2.5) / MAX (8.67, 2.5) = 6.17/8.67 = 0.71

Calculate similarly SC(A2), SC(A3)

Then Calculate SC(k1) = [SC(A1) + SC(A2) + SC(A3)] / 3

Calculate similarly SC(k2), SC(k3)

Now Calculate SC(D) = [SC(k1) + SC(k2) + SC(k3)] / 3

▼ *12. **Supervised Learning > K-NN and** Weighted K-NN*

Sample	F1	F2	Class
s1	1	2	c1
s2	1	3	c1
s3	5	1	c2
s4	5	3	c2
s5	2	3	c1

Find out s6(3, 3) belong to which class

⇒

use **Euclidean Distance**

d(s6, s1) = (2^2 + 1^2)^(1/2) = **2.23**

d(s6, s2) = (2^2 + 0^2)^(1/2) = 2

d(s6, s3) = (2^2 + 2^2)^(1/2) = **2.82**

d(s6, s4) = (2^2 + 0^2)^(1/2) = 2

d(s6, s5) = (1^2 + 0^2)^(1/2) = 1

We can Calculate in 2 way

1. Normal

if k=1

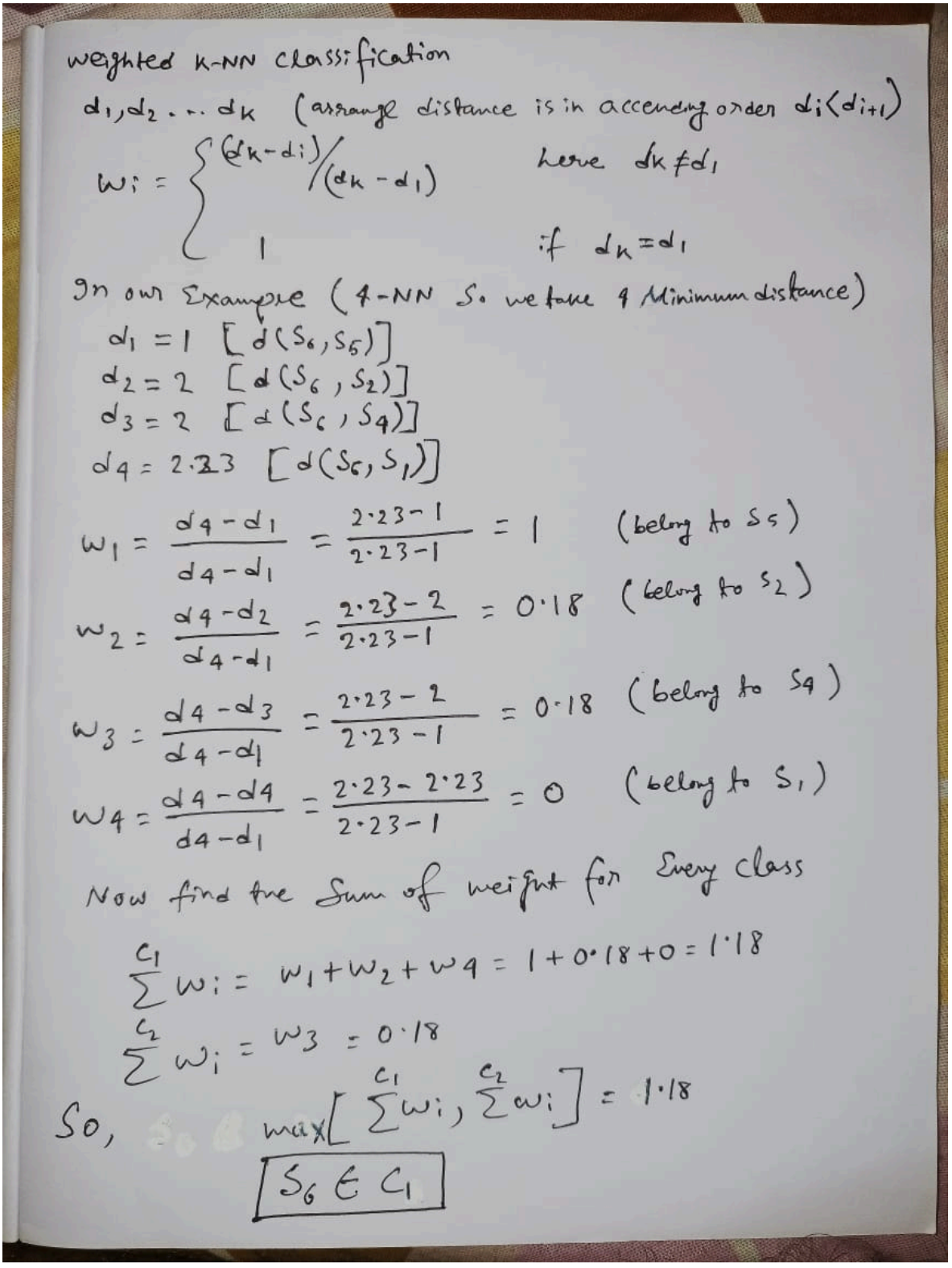
S6 belongs to c1 (d(s6, s5) minimum distance, and S5 belongs to c1)

if k=3

take minimum 3 then count c1, c2

- don't take even number of k, it occurs conflict

2. Weighted K-NN Classification



▼ *13.* ***Supervised > Classification >*** *Decision Tree > ID3*

Decision Tree (ID3 Algorithm) (from Jake Smashers)

Day	f_1	f_2	f_3	f_4	Class
1	S	H	H	W	N
2	S	H	H	S	N
3	C	H	H	W	Y
4	R	M	H	W	Y
5	R	C	N	W	Y
6	R	C	N	S	N
7	C	C	N	S	Y
8	S	M	H	W	N
9	S	C	N	W	Y
10	R	M	N	W	Y
11	S	M	N	S	Y
12	C	M	H	S	Y
13	C	H	N	W	Y
14	R	M	H	S	N

f_1 or weather $\Rightarrow$ {Sunny, Cloudy, Rain}
f_2 or Temperature $\Rightarrow$ {Hot, mild, Cold}
f_3 or Humidity $\Rightarrow$ {High, Normal}
f_4 or wind $\Rightarrow$ {Strong, weak}
class = play football = {Yes, No}

E : Entropy
IG : Information Gain
SI : Split Information

$\Rightarrow$ Calculate $IG(D)$ of (f_1)

Step 1: Entropy of Entire data set

Number of N = 5
" " Y = 9

$$E(D) = -\sum_{i=1}^{K} p_i \log_2(p_i)$$

$$= -\frac{9}{14}\log_2(9/14) - \frac{5}{14}\log_2(5/14)$$

$$= 0.94$$

Step 2: Entropy of all the attributes

5 when Y = 2
5 when N = 3

$$E(D_i) = -\sum_{j=1}^{K} p_j \log_2 p_j$$

$$E(S)\{+2,-3\} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$$

$$E(C)\{+4,-0\} = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

$$E(R)\{+3,-2\} = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$$

Here $E(f_1, D) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} E(D_i)$

$$E(f_1, D) = \frac{5}{14}E(S) + \frac{4}{14}E(C) + \frac{5}{14}E(R)$$

$$= \frac{5 \times 0.97}{14} + 0 + \frac{5 \times 0.97}{14}$$

$$= 0.692$$

$$IG(f_1) = E(D) - E(f_1, D)$$

$$= 0.94 - 0.692 = 0.247$$

Calculate IG of (f_2)

step1: $E(D) = 0.94$ (one time Calculation)

step2: Entropy of all the attributes,

$E(H)\{+2,-2\} = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$

$E(M)\{+4,-2\} = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.91$

$E(C)\{+3,-1\} = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.81$

$E(f_2, D) = \frac{4}{14}\times 1 + \frac{6}{14}\times 0.91 + \frac{4}{14}\times 0.81 = 0.9$

$IG(f_2) = E(D) - E(f_2, D) = 0.94 - 0.9 = 0.04$

Calculate IG of (f_3)

step1 : $E(D) = 0.94$

step2: Entropy of all the attributes

$E(H)\{+3,-4\} = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log\frac{4}{7} = 0.98$

$E(N)\{+6,-1\} = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log\frac{1}{7} = 0.59$

$E(f_3, D) = \frac{7\times 0.98}{14} + \frac{7\times 0.59}{14} = 0.785$

$IG(f_3) = E(D) - E(f_3, D) = 0.94 - 0.785 = 0.155$

Calculate IG of (f_4)

step1: $E(D) = 0.94$

step2: Entropy of all the attributes

$E(S)\{+3,-3\} = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$

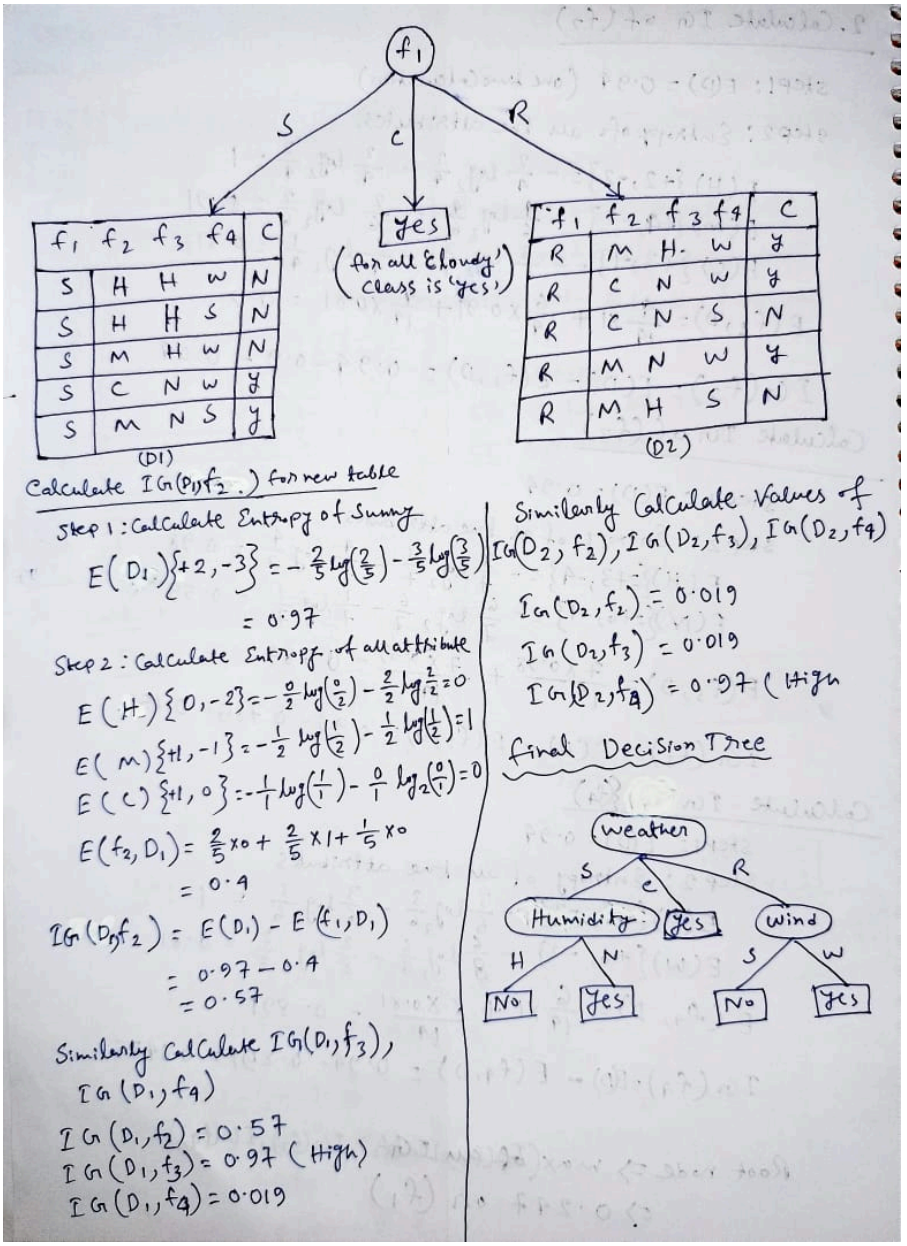
$E(W)\{+6,-2\} = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} = 0.81$

$E(f_4, D) = \frac{6}{14}\times 1 + \frac{8\times 0.81}{14} = 0.891$

$IG(f_4) = E(D) - E(f_4, D) = 0.94 - 0.891 = 0.0485$

Root node $\Rightarrow \max(IG(f_1), IG(f_2), IG(f_3), IG(f_4))$

$\Rightarrow 0.247$ or (f_1)



Advantages of ID3:

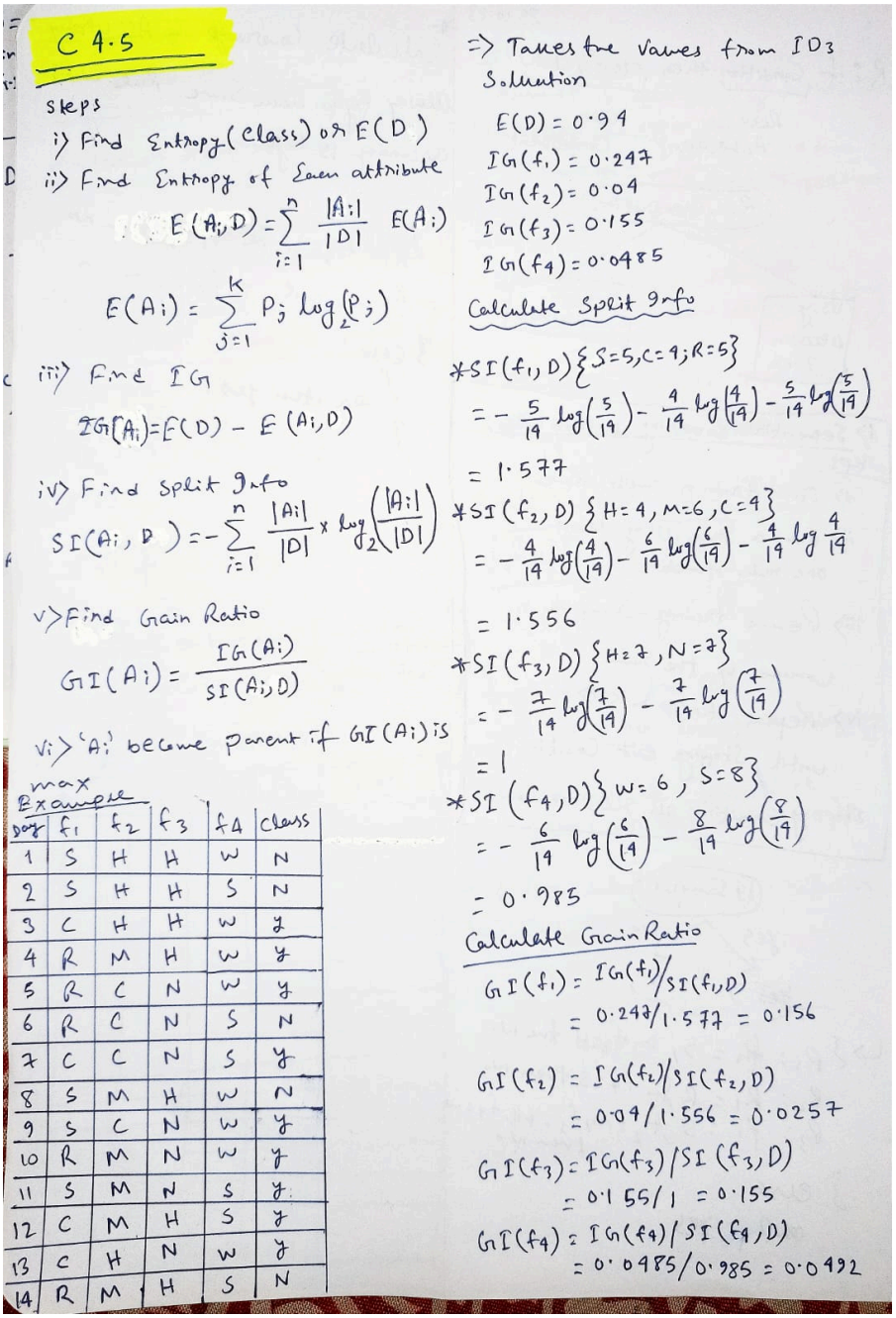
1. **Simplicity:** ID3 is relatively simple to understand and implement, making it accessible for those new to decision tree algorithms.
2. **Interpretability:** The resulting decision tree is easy to interpret, as it can be represented graphically and in a way that is understandable to non-experts.
3. **Feature Selection:** ID3 can automatically select important features, helping to identify the most relevant attributes for classification.
4. **Categorical Data:** ID3 is well-suited for datasets with categorical (discrete) features since it was primarily designed for such data.

Disadvantages of ID3:

1. **Binary Trees:** ID3 generates binary trees, meaning each node has only two branches. This can lead to relatively shallow trees and may not handle complex relationships well.
2. **Attribute Bias:** ID3 tends to favor attributes with more values because it selects attributes based on information gain, which is influenced by the number of possible values an attribute can take.

3. **Overfitting:** ID3 is prone to overfitting, especially with small datasets. The tree can become too specific to the training data and perform poorly on unseen data.
4. **Lack of Pruning:** ID3 does not include a pruning mechanism to avoid overfitting, which is a significant disadvantage compared to more modern decision tree algorithms like C4.5 and CART.
5. **Handling Missing Data:** ID3 doesn't handle missing data well. It may exclude instances with missing values, which can result in a loss of information.
6. **Numeric Data:** ID3 is not designed to handle continuous (numeric) data directly, requiring discretization or binning of continuous attributes.

▼ 14. ***Supervised > Classification >*** *Decision Tree > C4.5*



C 4.5

Steps

i) Find Entropy (Class) or E(D)

ii) Find Entropy of Each attribute

$$E(A_i, D) = \sum_{i=1}^{n} \frac{|A_i|}{|D|} E(A_i)$$

$$E(A_i) = \sum_{j=1}^{k} P_j \log_2(P_j)$$

iii) Find IG

$$IG(A_i) = E(D) - E(A_i, D)$$

iv) Find Split Info

$$SI(A_i, D) = -\sum_{i=1}^{n} \frac{|A_i|}{|D|} \times \log_2\left(\frac{|A_i|}{|D|}\right)$$

v) Find Gain Ratio

$$GI(A_i) = \frac{IG(A_i)}{SI(A_i, D)}$$

vi) 'Ai' become parent if GI(Ai) is max

Example

Day	f_1	f_2	f_3	f_4	class
1	S	H	H	W	N
2	S	H	H	S	N
3	C	H	H	W	Y
4	R	M	H	W	Y
5	R	C	N	W	Y
6	R	C	N	S	N
7	C	C	N	S	Y
8	S	M	H	W	N
9	S	C	N	W	Y
10	R	M	N	W	Y
11	S	M	N	S	Y
12	C	M	H	S	Y
13	C	H	N	W	Y
14	R	M	H	S	N

=> Takes the values from ID3 Solution

E(D) = 0.94

IG(f1) = 0.247

IG(f2) = 0.04

IG(f3) = 0.155

IG(f4) = 0.0485

Calculate Split Info

*SI(f1, D){S=5, C=4, R=5}

$= -\frac{5}{14}\log\left(\frac{5}{14}\right) - \frac{4}{14}\log\left(\frac{4}{14}\right) - \frac{5}{14}\log\left(\frac{5}{14}\right)$

= 1.577

*SI(f2, D){H=4, M=6, C=4}

$= -\frac{4}{14}\log\left(\frac{4}{14}\right) - \frac{6}{14}\log\left(\frac{6}{14}\right) - \frac{4}{14}\log\frac{4}{14}$

= 1.556

*SI(f3, D){H=7, N=7}

$= -\frac{7}{14}\log\left(\frac{7}{14}\right) - \frac{7}{14}\log\left(\frac{7}{14}\right)$

= 1

*SI(f4, D){W=6, S=8}

$= -\frac{6}{14}\log\left(\frac{6}{14}\right) - \frac{8}{14}\log\left(\frac{8}{14}\right)$

= 0.985

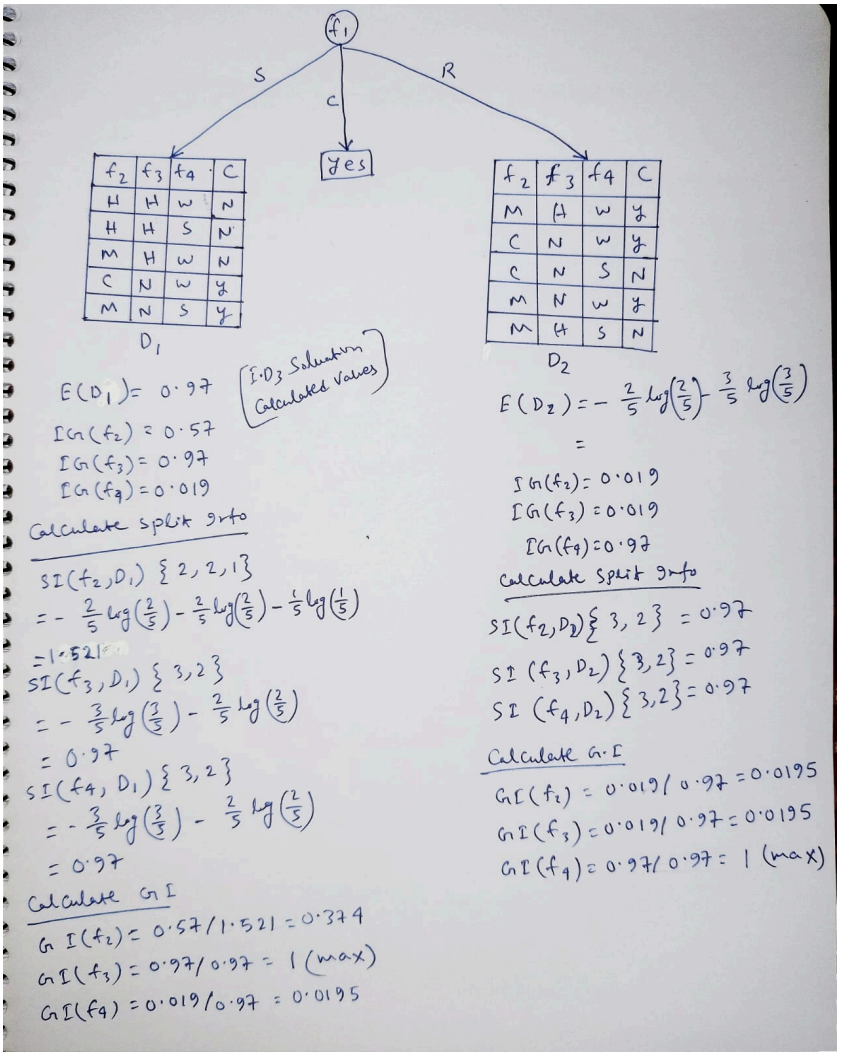
Calculate Gain Ratio

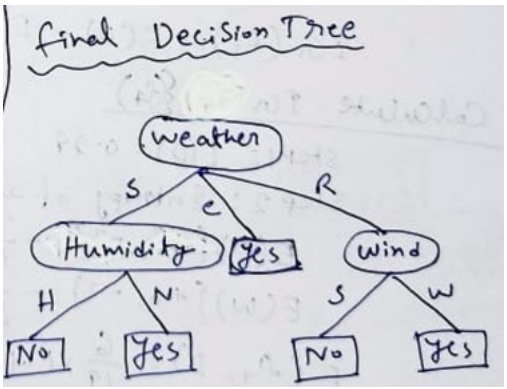
GI(f1) = IG(f1)/SI(f1, D)
= 0.247/1.577 = 0.156

GI(f2) = IG(f2)/SI(f2, D)
= 0.04/1.556 = 0.0257

GI(f3) = IG(f3)/SI(f3, D)
= 0.155/1 = 0.155

GI(f4) = IG(f4)/SI(f4, D)
= 0.0485/0.985 = 0.0492





▼ *15.* ***Supervised > Classification >*** *Decision Tree > CART*

Decision Tree using CART Algorithm

Day	f_1	f_2	f_3	f_4	Class
1	S	H	H	W	N
2	S	H	H	S	N
3	C	H	H	W	Y
4	R	M	H	W	Y
5	R	C	N	W	Y
6	R	C	N	S	N
7	C	C	N	S	Y
8	S	M	H	W	N
9	S	C	N	W	Y
10	R	M	N	W	Y
11	S	M	N	S	Y
12	C	M	H	S	Y
13	C	H	N	W	Y
14	R	M	H	S	N

(D)

=>

f_1	S	5	Y	2
			N	3
	C	4	Y	4
			N	0
	R	5	Y	3
			N	2
f_2	H	4	Y	2
			N	2
	M	6	Y	4
			N	2
	C	4	Y	3
			N	1
f_3	H	7	Y	3
			N	4
	N	7	Y	6
			N	1
f_4	W	6	Y	3
			N	3
	S	8	Y	6
			N	2

* we are looking for binary tree

number of strict binary tree $= 2^{K-1} - 1$

here K is number of attribute

Calculate G.I (Gini Index) of D

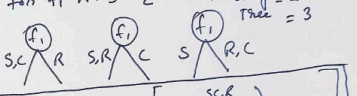
$$GI = 1 - \sum_{i=1}^{n} P_i^2$$

$$G.I(D) = 1 - \left[\left(\frac{9}{14}\right)^2 + \left(\frac{5}{14}\right)^2\right]$$

$$= 0.46$$

Calculate $G.I(f_1)$

for f_1 $K = 3$ $\{S, C, R\}$ strict binary tree $= 2^{3-1} - 1 = 3$



$G.I(f_1, D) = \min$ [$G.I(f_1^{SC,R}, D)$, $G.I(f_1^{SR,C}, D)$, $G.I(f_1^{S,RC}, D)$]

* we take min to get max G.I(f_1)

$$G.I(f_1^{SC,R}, D) = \frac{5+9}{14}\left[1 - \left(\frac{2+4}{9}\right)^2 - \left(\frac{3+0}{9}\right)^2\right] + \frac{5}{14}\left[1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2\right]$$

$$= 0.285 + 0.171$$

$$= 0.456$$

$$G.I(f_1^{SR,C}, D) = \frac{10}{14}\left[1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2\right] + \frac{4}{14}\left[1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2\right]$$

$$= 0.357 + 0 = 0.357$$

$$G.I(f_1^{S,RC}, D) = \frac{5}{14}\left[1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2\right] + \frac{9}{14}\left[1 - \left(\frac{7}{9}\right)^2 - \left(\frac{2}{9}\right)^2\right]$$

$$= 0.17 + 0.22 = 0.392$$

$G\cdot I(f_1, D) = 0.357 \quad \{S R, C\}$

$G\cdot I(f_1) = G\cdot I(D) - G\cdot I(f_1, D)$

$= 0.46 - 0.357$

$= 0.103$

Calculate $G\cdot I(f_2)$

$$G\cdot I(f_2, D) = \min \begin{bmatrix} G\cdot I(f_2^{H,M,C}, D), \\ G\cdot I(f_2^{HC,M}, D), \\ G\cdot I(f_2^{H,MC}, D) \end{bmatrix}$$

$$G\cdot I\left(f_2^{H,M,C}, D\right) = \frac{10}{14}\left[1-\left(\frac{6}{10}\right)^2-\left(\frac{4}{10}\right)^2\right] + \frac{4}{14}\left[1-\left(\frac{3}{4}\right)^2-\left(\frac{1}{4}\right)^2\right]$$

$= 0.342 + 0.107 = 0.449$

$$G\cdot I\left(f_2^{HC,M}, D\right) = \frac{8}{14}\left[1-\left(\frac{5}{8}\right)^2-\left(\frac{3}{8}\right)^2\right] + \frac{6}{14}\left[1-\left(\frac{4}{6}\right)^2-\left(\frac{2}{6}\right)^2\right]$$

$= 0.267 + 0.190 = 0.457$

$$G\cdot I\left(f_2^{H,MC}, D\right) = \frac{4}{14}\left[1-\left(\frac{2}{4}\right)^2-\left(\frac{2}{4}\right)^2\right] + \frac{10}{14}\left[1-\left(\frac{7}{10}\right)^2-\left(\frac{3}{10}\right)^2\right]$$

$= 0.125 + 0.3 = 0.425$

$G\cdot I(f_2, D) = 0.425 \quad \{H, MC\}$

$G\cdot I(f_2) = G\cdot I(D) - G\cdot I(f_2, D)$

$= 0.46 - 0.425$

$= 0.035$

Calculate $G\cdot I(f_3)$

$$G\cdot I(f_3, D) = \frac{7}{14}\left[1-\left(\frac{3}{7}\right)^2-\left(\frac{4}{7}\right)^2\right] + \frac{7}{14}\left[1-\left(\frac{6}{7}\right)^2-\left(\frac{1}{7}\right)^2\right]$$

$= 0.244 + 0.122 = 0.366$

$G\cdot I(f_3) = G\cdot I(D) - G\cdot I(f_3, D) = 0.46 - 0.366 = 0.094$

Calculate $G\cdot I(f_4)$

$$G\cdot I(f_4, D) = \frac{6}{14}\left[1-\left(\frac{3}{6}\right)^2-\left(\frac{3}{6}\right)^2\right] + \frac{8}{14}\left[1-\left(\frac{6}{8}\right)^2-\left(\frac{2}{8}\right)^2\right]$$

$= 0.214 + 0.214 = 0.428$

$G\cdot I(f_4) = G\cdot I(D) - G\cdot I(f_4, D) = 0.46 - 0.428 = 0.032$

$G\cdot I$ for 1st iteration

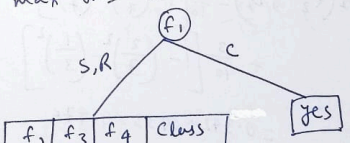
$G\cdot I(f_1) = 0.103$

$G\cdot I(f_2) = 0.035$

$G\cdot I(f_3) = 0.094$

$G\cdot I(f_4) = 0.032$

max $G\cdot I$ is $G\cdot I(f_4)$



f_2	f_3	f_4	Class
H	H	W	N
H	H	S	N
M	H	W	Y
C	N	W	Y
C	N	S	N
M	H	W	N
C	N	W	Y
M	N	W	Y
M	N	S	Y
M	H	S	N

(D_1)

2nd iteration

Calculate G.I for D_1

$$G.I(D_1) = 1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2 = 0.5$$

		Total	yes	No
f_2	H	2	0	2
	M	5	3	2
	C	3	2	1
f_3	H	5	1	4
	N	5	4	1
f_4	W	6	4	2
	S	4	1	3

Calculate $G.I(f_2)$

$$G.I(f_2, D_1) = \min\begin{bmatrix} G.I(f_2^{H,M,C}, D_1), \\ G.I(f_2^{H,C,M}, D_1), \\ G.I(f_2^{H,CM}, D_1) \end{bmatrix}$$

$$G.I\left(f_2^{H,M,C}, D_1\right) = \frac{7}{10}\left[1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2\right] + \frac{3}{10}\left[1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2\right]$$

$$= 0.342 + 0133 = 0.475$$

$$G.I\left(f_2^{H,C,M}, D_1\right) = \frac{5}{10}\left[1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2\right] + \frac{5}{10}\left[1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2\right]$$

$$= 0.24 + 0.24 = 0.48$$

$$G.I\left(f_2^{H,CM}, D_1\right) = \frac{2}{10}\left[1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2\right] + \frac{8}{10}\left[1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2\right]$$

$$= 0 + 0.375 = 0.375$$

$$GI(f_2, D_1) = 0.375$$

$$GI(f_2) = 0.5 - 0.375 = 0.125$$

Calculate $GI(f_3)$

$$GI(f_3, D_1) = \frac{5}{10}\left[1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2\right] + \frac{5}{10}\left[1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2\right]$$

$$= 0.16 + 0.16 = 0.32$$

$$GI(f_3) = 0.5 - 0.32 = 0.18$$

Calculate $GI(f_4)$

$$GI(f_4, D_1) = \frac{6}{10}\left[1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2\right] + \frac{4}{10}\left[1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right]$$

$$= 0.266 + 0.15 = 0.416$$

$$GI(f_4) = 0.5 - 0.416 = 0.084$$

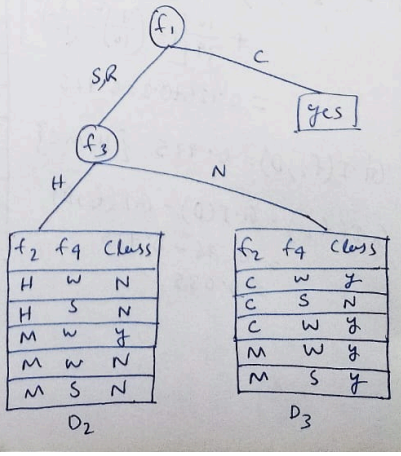
GI for 2nd iteration

$GI(f_2) = 0.125$

$GI(f_3) = 0.18$

$GI(f_4) = 0.084$

Max GI is $GI(f_3)$

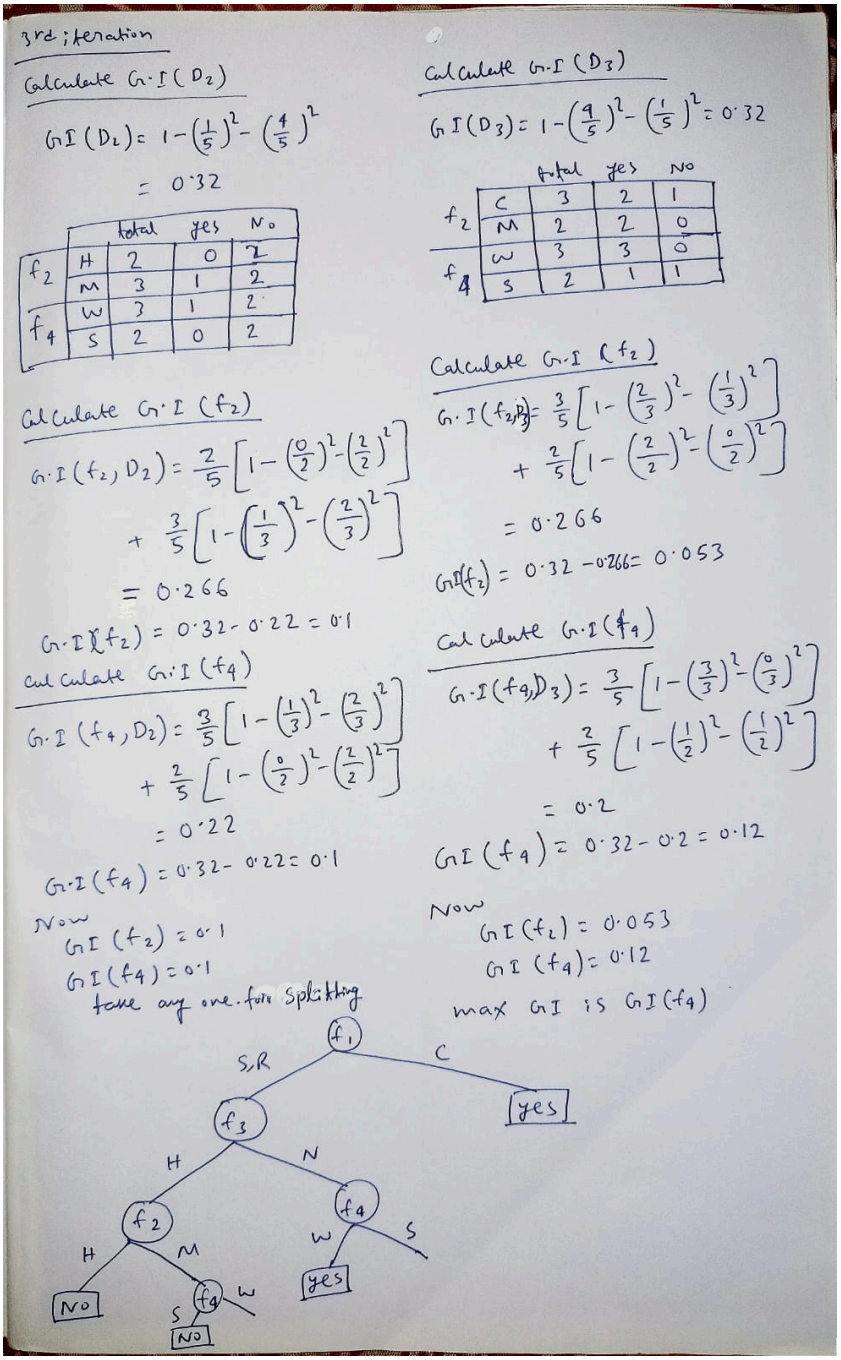


f_2	f_4	Class
H	W	N
H	S	N
M	W	Y
M	W	N
M	S	N

D_2

f_2	f_4	Class
C	W	Y
C	S	N
C	W	Y
M	W	Y
M	S	Y

D_3



▼ 16. *Supervised > Classification > Rule Based >Sequential Covering Algorithm*

https://web.iitd.ac.in/~bspanda/rb.pdf

https://www.tutorialspoint.com/data_mining/dm_rbc.htm

https://www3.cs.stonybrook.edu/~cse352/L8DTIntro.pdf

Rule-based classification in data mining is a technique in which the classification decisions are taken based on various IF-THEN rules.

Each rule of the classifier consists of an antecedent and a consequent.

Structure of a rule

IF condition THEN conclusion

or, condition → c

Ex :

1. IF age = youth AND student = yes THEN buy_computer = yes

2. (age = youth) ^ (student = yes)) (buys computer = yes)
3. (age = youth) ^ (student = yes) → (buys computer = yes)

Rule Antecedent or Precondition or Condition : The Left Hand Side("IF" part) of a rule is called the rule antecedent or condition.

The antecedent may have one or more conditions, which are logically ANDed.

Rule Consequent or Condition : The Right Hand Side("THEN" part) of a rule is called the rule consequent.

Rule consequent consists of class prediction. The class prediction is the leaf node or end node.

Assessment of Rule

Rule can be accessed based on two factors. Let's define a few parameters first.

na = number of records covered by the rule(R).

nc = number of records correctly classified by rule(R).

n = Total number of records

Coverage of a rule: Fraction of records that satisfy the rule's antecedent describes rule coverage.

```
Coverage (R) = na  / |n|
```

Accuracy of a rule: Fraction of records that meet the antecedent and consequent value defines rule accuracy.

```
Accuracy (R) = nc / na
```

- We can convert the ***Assessment*** result into percentage by multiplying them by 100.
- One sample should be trigger only one rule.
- Our target is to reduce the rule set and cover all coverage and accuracy.
- Always try to make some rule who's accuracy is high.

Sequential Covering Algorithm

1. Start from a empty Rule
2. Make a new rule, using learn. one-rule-function.
3. Remove training Sample Covered by the rule.
4. Repeat step (2) & (3). until stopping Condition.
5. Stopping Condition : All rules are Covered.

Rule Generation

Indirect Method : we extract rules from different other classification models. Some prominent classifications from which we extract the rules are decision trees and neural networks.

EX : Decision trees

Direct Method : The direct method for rule-based classification in data mining contains algorithms that extract rules directly from the dataset.

EX : 1R Algorithm, Sequentia Covering Algorithm

Properties of Rule Based Data Mining Classifiers

There are two significant properties of rule-based classification in data mining. They are:

- **Rules may not be mutually exclusive**
- **Rules may not be exhaustive**

1. ***Rules may not be mutually exclusive***

Many different rules are generated for the dataset, so it is possible and likely that many of them satisfy the same data record. This condition makes the rules not mutually exclusive.

Since the rules are not mutually exclusive, we cannot decide on classes that cover different parts of data on different rules. But this was our main objective. So, to solve this problem, we have two ways:

- **Ordered rule set**: Rank the rules according to their priority, and the class corresponding to the highest-ranked rule is taken as the final class

First select the class which has cover less number of rules.

By ordering the rules, we set priority orders.

Thus, this ordered rule set is called a decision list.

So the class with the highest priority rule is taken as the final class.

- **Unordered rule set**: Votes are assigned to each class depending on their weights

2. **Rules may not be exhaustive**

It is not a guarantee that the rule will cover all the data entries.

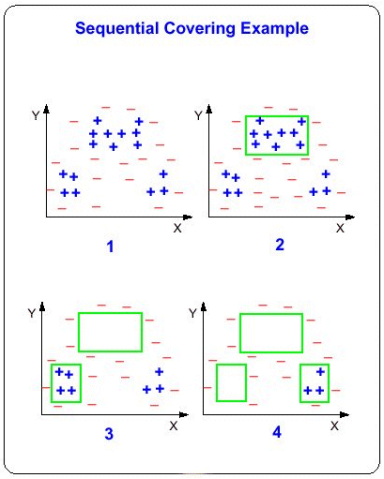
So, to solve this problem, we can make ***use of a default class.***

Using a default class, we can assign all the data entries not covered by any rules to the default class. Thus using the default class will solve the problem of non-exhaustivity.

- Decision tree satisfy *mutually exclusive and* exhaustive

Advantages of Rule-Based Classification

- Highly expressive.
- Easy to interpret.
- Easy to generate.
- Capability to classify new records rapidly.
- Performance is comparable to other classifiers



Rule-based Classifier (Example 1) for Vertebrate Classification Problem

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1 : (Give Birth = no) and (Can Fly = yes) → Birds
R2 : (Give Birth = no) and (Live in Water = yes) → Fishes
R3 : (Give Birth = yes) and (Blood Type = warm) → Mammals
R4 : (Give Birth = no) and (Can Fly = no) → Reptiles
R5 : (Live in Water = sometimes) → Amphibians

Testing

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

The rule R1 covers a hawk ⇒ Bird

The rule R3 covers the grizzly bear ⇒ Mammal

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Example 2

Age	Income	Credit Score	Class
25	20,000	700	Approved
30	30,000	800	Approved
35	40,000	900	Approved
20	15,000	600	Denied
22	18,000	650	Denied

Goal : Create a set of rules to predict whether a loan application will be approved or denied.

Steps:

1. **Initialize an empty rule set.**
2. **Learn one rule to cover as many positive examples as possible.**
   - **Rule 1:** IF Age >= 30 AND Income >= 20,000 THEN Approved
   - This rule covers 3 positive examples (ages 30, 35, and 40) and 0 negative examples.
3. **Remove the examples covered by the rule.**
4. **Learn another rule to cover as many positive examples as possible.**
   - **Rule 2:** IF Age >= 25 AND Credit Score >= 700 THEN Approved
   - This rule covers 2 positive examples (ages 25 and 30) and 0 negative examples.
5. **Remove the examples covered by the rule.**
6. **Repeat steps 2-5 until all positive examples are covered.**
   - **Rule 3:** IF Age >= 20 AND Credit Score >= 650 THEN Approved
   - This rule covers 1 positive example (age 20) and 0 negative examples.

Final rule set:

1. IF Age >= 30 AND Income >= 20,000 THEN Approved
2. IF Age >= 25 AND Credit Score >= 700 THEN Approved
3. IF Age >= 20 AND Credit Score >= 650 THEN Approved

This rule set can be used to classify new loan applications. For example, if a new applicant is 27 years old with an income of $25,000 and a credit score of 750, the rule set would predict that their application would be approved.

Coverage:

- Rule 1: 3/5 = 0.60
- Rule 2: 2/5 = 0.40
- Rule 3: 1/5 = 0.20

Overall coverage: (3 + 2 + 1)/5 = 6/5 = 1.20

Since overall coverage is greater than 1, it means that some examples are covered by multiple rules. This is because the rules are not mutually exclusive.

Accuracy:

- Rule 1: 3/3 = 1.00
- Rule 2: 2/2 = 1.00
- Rule 3: 1/1 = 1.00

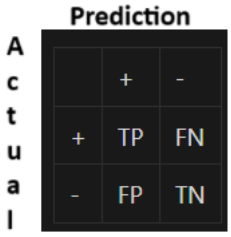
Overall accuracy: (3 + 2 + 1)/(3 + 2) = 6/5 = 1.20

Since overall accuracy is greater than 1, it means that some examples are correctly classified by multiple rules. This is because the rules are not mutually exclusive.

▼ *17. Performance Measure for Supervised Learning*

For Design the model we use 70% of total data set and 30% for testing.

1. ***Confusion Matrix***



TP : True Positive

FN : False Negative

FP : False Positive

TN : True Negative

Total Testing Sample : TP+FN+FP+TN

2. Accuracy = (TP+ TN)/Total Sample
3. Error = 1 - Accuracy
4. True Positive Rate (TPR) or **Recall or Sensitivity** = TP / (TP+FN)
5. True Negative Rate (TNR) or **Specificity** = TN / (FP+TN)
6. False Positive Rate (FPR) or Type 1 Error = FP / (FP+TN) = 1 - TPR
7. False Negative Rate (FNR) or Type 2 Error = FN / (TP+FN) = 1 - TNR
8. **Precision = TP/(TP+FP)**

Precision+ = TP/(TP+FP)

Precision+ = TN/(TN+FN)

9. **Precision Error(PE)**

PE+ = FP/(FP+TP)

PE - = FN/(FN+TN)

10. **Fβ Measurement=(1+β^2)*P*R / (P*β^2 + R), β should be non (-ve) real number**

Types Of **Fβ Measurement**

	F0.5 Score = 3*P*R/(P+4R)
If P & R both important then β=1	**F1 Score = 2*P*R/(P+R)**
Increase Accuracy β=2	**F2 Score = 5*P*R/(4*P+R)**

Example

need some example

Performance measurement for multiclass classification.

Actual \ predicted	C_1	C_2	C_3	C_4
C_1	50	5	2	3
C_2	9	70	11	0
C_3	7	10	30	3
C_4	30	48	2	60

Actual \ predicted	+	−
+	TP	FN
−	FP	TN

Number of Sample = 50+5+2+3 +9+70+11+0 +7+10+30+3 +30+48+2+60 = 340

$TP_{C_1} = 50$ $TP_{C_3} = 30$

$TP_{C_2} = 70$ $TP_{C_4} = 60$

TP = diagonal Values $= TP_{C_1} + TP_{C_2} + TP_{C_3} + TP_{C_4}$
$= 50 + 70 + 30 + 60$
$= 210$

$FN_{C_1} = 5+2+3 = 10$
$FN_{C_2} = 9+11+0 = 20$
$FN_{C_3} = 7+10+3 = 20$
$FN_{C_4} = 30+48+2 = 80$

$FP_{C_1} = 9+7+30 = 46$
$FP_{C_2} = 5+10+48 = 63$
$FP_{C_3} = 2+11+2 = 15$
$FP_{C_4} = 3+0+3 = 6$

$$TN_{C_1} = \begin{pmatrix} 70 & 11 & 0 \\ 10 & 30 & 3 \\ 48 & 2 & 60 \end{pmatrix} = \begin{matrix} 70+11+0+30+3 \\ +48+2+60 \end{matrix} = 234$$

OR $= Total - TP_{C_1} - FN_{C_1} - FP_{C_1}$
$= 340 - 50 - 10 - 46$
$= 234$

$TN_{C_2} = 340 - 70 - 20 - 63 = 187$
$TN_{C_3} = 340 - 30 - 20 - 15 = 275$
$TN_{C_4} = 340 - 60 - 80 - 6 = 194$

Accuracy $= \frac{TP + TN}{Total}$ neglect 'TN'

$= \frac{210}{340}$

$= 0.6176 = 61.76\%$

Error $= 1 - 0.6176 = 0.3823$
$= 38.23\%$

Similarly Calculate other Values also

$TPR_{C_1} = TP_{C_1} / TP_{C_1} + FN_{C_1}$
$= 50 / 50+10 = \frac{50}{60} = 0.833$

Performance Measure for probability Score

Sample	Actual class	Predicted Probability
S_1	+	0.6
S_2	+	0.8
S_3	−	0.7
S_4	+	0.5
S_5	+	0.55
S_6	−	0.54
S_7	−	0.4
S_8	−	0.51
S_9	+	0.9
S_{10}	−	0.53

* here we don't have exact prediction
* we have predicted data in Probability
* we have to find out a 'threshold' which gives us maximum Accuracy.

Steps (i) set 'threshold' high to low value from probability Column.
(ii) Find out accuracy from Confusion matrix.
(iii) obtain max accuracy & select that threshold.

here, if S_i value $\geq$ threshold
then '+' class
else '−' class

* set different threshold & Calculate Accuracy

Sample	Actual	Prob	t_1 0.9	t_2 0.8	t_3 0.7	t_4 0.6	t_5 0.55	t_6 0.54	t_7 0.53	t_8 0.51	t_9 0.5	t_{10} 0.4
S_9	+	0.9	+	+	+	+	+	+	+	+	+	+
S_2	+	0.8	−	+	+	+	+	+	+	+	+	+
S_3	−	0.7	−	−	+	+	+	+	+	+	+	+
S_1	+	0.6	−	−	−	+	+	+	+	+	+	+
S_5	+	0.55	−	−	−	−	+	+	+	+	+	+
S_6	−	0.54	−	−	−	−	−	+	+	+	+	+
S_{10}	−	0.53	−	−	−	−	−	−	+	+	+	+
S_8	−	0.51	−	−	−	−	−	−	−	+	+	+
S_4	+	0.5	−	−	−	−	−	−	−	−	+	+
S_7	−	0.4	−	−	−	−	−	−	−	−	−	+
TP			1	2	2	3	4	4	4	4	5	5
TN			5	5	4	4	4	3	2	1	1	0
Total			6	7	6	7	8	7	6	5	6	5
Accuracy			0.6	0.7	0.6	0.7	0.8	0.7	0.6	0.5	0.6	0.5

when $t_1 = 0.9$

Actual \ predicted	+	−
+	1 TP	4 FN
−	0 FP	5 TN

Accuracy $= \frac{TP + TN}{total}$

$= \frac{6}{10} = 0.6$

we get max Accuracy when Threshold = 0.55
Then Accuracy = 0.8

▼ 18. *Data Mining* Introduction

Data analysis like Data visualization

Data analysis is the process of collecting, cleaning, and transforming data to extract meaningful insights. It involves a variety of techniques, including:

1. **Descriptive statistics:** Summarize the data using measures like mean, median, mode, and standard deviation.
2. **Inferential statistics:** Draw conclusions about the population based on a sample of data.
3. **Data mining:** Uncover hidden patterns and trends in large datasets.
4. **Machine learning:** Build models that can make predictions or decisions based on data.

Feature	Data Analysis	Data Visualization
Focus	Collecting, cleaning, and transforming data	Representing data in a visual format
Goals	Extract meaningful insights from data	Help people understand and communicate data
Techniques	Descriptive statistics, inferential statistics, data mining, machine learning	Bar charts, line charts, pie charts, scatter plots, maps
Output	Tables, reports, models	Charts, graphs, maps

Data Mining and Knowledge Discovery in Databases

Data Mining : The process of extracting implicit, previously unknown, and potentially useful information from data in databases.

Knowledge Discovery in Databases (KDD) : A broader process that includes data mining as well as the steps of data selection, data preprocessing, data transformation, and data interpretation/evaluation.

KDD Process :

1. Selection: Create a target dataset or focus on a subset of variables or data samples on which discovery is to be performed.

2. Preprocessing: Clean and preprocess the target data in order to obtain consistent data. **3. Transformation:** Transform the data using dimensionality reduction or transformation methods.

4. Data Mining: Search for patterns of interest in a particular representational form, depending on the DM objective (usually prediction).

5. Interpretation/Evaluation: Interpret and evaluate the mined patterns.

Data Mining Tasks	Classification, prediction, association rule mining, clustering, outlier detection
Data Mining Algorithms	Decision trees, neural networks, support vector machines, k-means clustering, DBSCAN
Data Mining Applications	Fraud detection, customer segmentation, risk assessment, medical diagnosis, scientific discovery

Data Mining Functionalities

Data Mining Functionality	Description	Example Algorithm
Data Cleaning	Identify and correct errors, inconsistencies, and missing values in the data.	Iterative outlier detection algorithm, data scrubbing
Data Integration	Combine and unify data from multiple sources into a single, consistent dataset.	Data wrangling, data reconciliation
Data Selection	Select the most relevant and informative data for further analysis.	Feature selection, sampling
Data Transformation	Convert or transform data into a format that is suitable for data mining algorithms.	Data normalization, dimensionality reduction

Data Mining	Discover patterns, trends, and relationships in the data.	Classification algorithms, association rule mining, clustering
Pattern Evaluation	Assess the quality and significance of the mined patterns.	Statistical tests, evaluation metrics
Knowledge Presentation	Communicate the discovered knowledge to stakeholders in a clear and understandable way.	Data visualization, report generation

Data Pre-Processing (Data Transformation)

Data preprocessing is a crucial step in the data analysis process that involves transforming and preparing raw data into a suitable format for further analysis. It aims to enhance the quality, reliability, and usability of the data before it is fed into machine learning algorithms, statistical models, or other analytical tools.

Data Transformation : Converting data into a format that is compatible with the chosen analysis tools and algorithms. This may involve scaling, normalization, discretization, or encoding.

Feature Engineering: Creating new features from existing ones or selecting a subset of relevant features to improve the performance of the analysis.

Data Preprocessing Step	Description
Data Cleaning	Identifying and correcting errors, inconsistencies, and missing values in the data.
Data Integration	Combining and unifying data from multiple sources into a single, consistent dataset.
Data Selection	Selecting the most relevant and informative data for further analysis.
Data Transformation	Converting or transforming data into a format that is suitable for data mining algorithms.
Data Reduction	Reducing the dimensionality of the data to make it more manageable and improve the performance of data mining algorithms.
Data Discretization	Converting continuous variables into discrete variables.
Data Normalization	Standardizing the values of numerical variables to a common range.
Data Encoding	Representing categorical variables in a way that is understandable to data mining algorithms.

B**enefits of data preprocessing:**

- **Improved quality of data:** Data preprocessing can help to identify and correct errors in the data, which can lead to more accurate results.
- **Increased efficiency of data mining algorithms:** Preprocessed data can be more easily processed by data mining algorithms, which can lead to better performance.
- **Improved interpretability of results:** Preprocessed data can be easier to understand and interpret, which can help to make the results of the analysis more meaningful.

Data Cleaning

Data cleaning, also known as data scrubbing, is the process of identifying and correcting errors, inconsistencies, and missing values in data. It is an important step in the data preprocessing phase of data mining, machine learning, and other data analysis tasks. By cleaning the data, we can ensure that it is accurate, complete, and consistent, which can lead to better results from our analyses

Steps are

Identifying missing values

Handling outliers

Correcting data entry errors

Resolving inconsistencies

Data Integration

Combining and unifying data from multiple sources into a single, consistent dataset.

Steps are

Data Identification

Data Acquisition

Data Transformation

Data Merging

Data Reconciliation

Data Validation

Data Reduction

Data reduction is the process of transforming or encoding data into a more compact representation while preserving the essential information or patterns within the data.

It aims to reduce the size of the data without significantly compromising its quality or usefulness.

Data reduction techniques are often employed in data mining, machine learning, and data compression applications.

Steps are

Dimensionality Reduction	Transforming the data into a lower-dimensional space while preserving as much of the information as possible. Example : PCA, SVD, LLE, Isomap
Feature Selection	Selecting a subset of the most relevant features from the original dataset.
Numerosity Reduction	Reducing the number of instances (records) in the dataset. Techniques include sampling methods, aggregation, and clustering.
Data Compression	Reducing data size through encoding or compression techniques. Examples include run-length encoding, Huffman coding, and other compression algorithms.
Binning or Histograms	Grouping continuous data into intervals (bins) to reduce the number of unique values. Common in numerical data representation.
Thresholding	Setting a threshold to filter out data points that fall below or above a certain value. Used in image processing and sensor data to reduce noise or focus on specific ranges of interest.

Data Discretization

Data discretization is defined as **a process of converting continuous data attribute values into a finite set of intervals and associating with each interval some specific data value**.

	Age
Before Discretization	1,5,9,4,7,11,14,17,13,18, 19,31,33,36,42,44,46,70,74,78,77
After Discretization	Child : 1, 5, 4, 9, 7 Young : 11, 14, 17, 13, 18, 19 Mature : 31, 33, 36, 42, 44, 46 Old : 70, 74, 77, 78

Concept Hierarchy Generation

The process of creating a hierarchical structure that represents relationships and dependencies between different levels of abstraction or granularity within a dataset.

Concept Hierarchy Generation Technique	Description
Top-down	Start with a single concept and then recursively divide it into more specific concepts until a stopping criterion is met. **Example Algorithms :** Hierarchical Agglomerative Clustering (HAC), Conceptual Clustering

Bottom-up	Start with a set of individual concepts and then iteratively merge them together to form higher-level concepts. **Example Algorithms :** C4.5, ID3

Data Mining Tasks

Data Mining Task	Description	Example Algorithms
Classification	Assigning a class label to a new data point based on its attributes.	Logistic Regression, Support Vector Machines (SVMs), Decision Trees
Prediction	Predicting a numerical value for a new data point based on its attributes.	Linear Regression, Random Forests, Neural Networks
Association Rule Mining	Discovering relationships between different attributes in a dataset.	Apriori Algorithm, FP-Growth Algorithm
Clustering	Grouping similar data points together based on their attributes.	K-Means Clustering, Hierarchical Clustering, DBSCAN
Outlier Detection	Identifying data points that are significantly different from the rest of the data.	Statistical Outlier Detection, Density-Based Outlier Detection
Pattern Recognition	Discovering recurring patterns in data.	Sequential Pattern Mining, Motif Discovery
Visualization	Representing data in a visual format to facilitate understanding and analysis.	Scatter Plots, Bar Charts, Line Charts

Data Mining Tools

Tool name	Description
RapidMiner	Open-source data mining platform with a graphical user interface and a wide range of algorithms and features.
SAS Enterprise Miner	Commercial data mining tool from SAS Institute.
SPSS Modeler	Statistical data analysis and data mining software from IBM.
Weka	Open-source machine learning toolkit developed at the University of Waikato in New Zealand.
KNIME	Open-source data mining and machine learning platform with a graphical user interface.
Orange	Open-source data visualization and machine learning toolkit.
TensorFlow	Open-source software library for numerical computation using data flow graphs.
Apache Spark MLlib	Open-source machine learning library built on top of Apache Spark.
Tableau	A business intelligence and data visualization tool that allows users to connect, visualize, and share data.
Power BI	A business analytics tool by Microsoft for interactive visualizations and business intelligence.
Oracle Data Mining	Part of the Oracle Advanced Analytics option, providing in-database data mining functionality.
IBM SPSS Modeler	A data mining and predictive analytics software from IBM that allows users to build predictive models without programming.

▼ 19. *Applications*

Healthcare

1. **Predictive Medicine**
2. **Personalized Medicine**
3. **Drug Discovery and Development**
4. **Fraud Detection and Abuse Prevention**
5. **Healthcare Resource Management**
6. **Predicting Hospital Readmissions**
7. **Identifying High-Risk Patients**
8. **Optimizing Treatment Plans**
9. **Detecting Adverse Drug Reactions**
10. **Improving Clinical Decision-Making**

Fraud detection

1. **Transaction Fraud Detection**
2. **Insurance Fraud Detection**
3. **Telecommunications Fraud Detection**
4. **Healthcare Fraud Detection**
5. **Tax Fraud Detection**

Common Data Mining Techniques for Fraud Detection:

1. **Clustering**
2. **Classification**
3. **Anomaly Detection**
4. **Association Rule Mining**
5. **Decision Trees**

Intrusion detection

Market basket analysis

Banking and Finance

Healthcare

Data mining has revolutionized the healthcare industry by enabling the extraction of valuable insights from vast amounts of medical data. This data, which includes patient records, electronic health records, medical imaging, and genomic data, holds the key to improving patient care, reducing healthcare costs, and advancing medical research.

Key Applications of Data Mining in Healthcare

1. **Predictive Medicine:** Data mining algorithms can analyze patient data to identify patterns and risk factors associated with various diseases. This allows for early detection and preventive measures, improving patient outcomes.
2. **Personalized Medicine:** Data mining can tailor treatment plans to individual patients based on their unique genetic, lifestyle, and medical history. This personalized approach leads to more effective and targeted treatments.
3. **Drug Discovery and Development:** Data mining can accelerate the drug discovery process by identifying potential drug candidates and predicting their efficacy and safety. This can save time and resources in drug development.

4. **Fraud Detection and Abuse Prevention:** Data mining can analyze insurance claims and healthcare transactions to identify patterns indicative of fraud or abuse. This helps to protect healthcare providers and insurers from financial losses.
5. **Healthcare Resource Management:** Data mining can optimize the allocation of healthcare resources, such as hospital beds, staff, and medical equipment, to improve efficiency and patient care.
6. **Predicting Hospital Readmissions:** Data mining can analyze patient data to predict the likelihood of rehospitalization, allowing for targeted interventions to prevent costly readmissions.
7. **Identifying High-Risk Patients:** Data mining can identify patients at high risk of developing chronic diseases, enabling early interventions and preventive measures to improve long-term health outcomes.
8. **Optimizing Treatment Plans:** Data mining can analyze patient data and treatment outcomes to identify the most effective treatment plans for specific conditions and patient populations.
9. **Detecting Adverse Drug Reactions:** Data mining can analyze patient data and pharmaceutical data to identify potential adverse drug reactions and improve drug safety.
10. **Improving Clinical Decision-Making:** Data mining can provide clinicians with real-time insights from patient data, supporting informed decision-making and improving patient care.

Conclusion

Data mining has become an indispensable tool in the healthcare industry, transforming patient care, medical research, and healthcare resource management. As the volume and complexity of healthcare data continue to grow, data mining will play an even more critical role in driving innovation and improving healthcare outcomes.

Fraud detection

Data mining plays a crucial role in fraud detection by enabling the identification of patterns and anomalies in large datasets that may indicate fraudulent activity. By leveraging data mining techniques, organizations can effectively detect and prevent fraud, minimizing financial losses and protecting their reputation.

Key Applications of Data Mining in Fraud Detection:

1. **Transaction Fraud Detection:** Data mining algorithms can analyze transaction data, such as credit card purchases or online transactions, to identify unusual patterns or deviations from normal spending behavior. This can help flag potential fraudulent transactions for further investigation.
2. **Insurance Fraud Detection:** Data mining techniques can analyze insurance claims data to detect anomalies or inconsistencies that may indicate fraudulent activity. This can help identify fraudulent claims before they are paid, reducing losses for insurance companies.
3. **Telecommunications Fraud Detection:** Data mining algorithms can analyze telecommunications usage data to identify suspicious patterns or anomalies, such as unusually high call volumes or international calls from unauthorized locations. This can help prevent fraudulent use of telecommunications services.
4. **Healthcare Fraud Detection:** Data mining techniques can analyze healthcare claims data to detect patterns that may indicate fraudulent activity, such as overbilling or duplicate claims. This can help prevent healthcare fraud and ensure that resources are allocated appropriately.
5. **Tax Fraud Detection:** Data mining algorithms can analyze tax return data to identify patterns that may indicate fraudulent activity, such as underreporting of income or excessive deductions. This can help tax authorities identify potential tax evaders and recover lost revenue.

Common Data Mining Techniques for Fraud Detection:

1. **Clustering:** Clustering algorithms group similar data points together, allowing for the identification of outliers or anomalies that may indicate fraudulent activity.
2. **Classification:** Classification algorithms can be trained on labeled data to distinguish between fraudulent and legitimate transactions or claims.
3. **Anomaly Detection:** Anomaly detection algorithms identify data points that deviate significantly from normal patterns, potentially signaling fraudulent activity.
4. **Association Rule Mining:** Association rule mining algorithms uncover relationships between data points, helping to identify patterns that may indicate fraudulent schemes.

5. **Decision Trees:** Decision trees provide a visual representation of decision-making processes, aiding in the understanding of fraud patterns and the development of fraud detection rules.

By employing data mining techniques, organizations can effectively combat fraud, safeguard their assets, and maintain their integrity. As fraudsters continue to develop sophisticated methods, data mining will remain an essential tool in the fight against financial crime.

Intrusion detection (Intrusion : অনুপ্রবেশ)

Data mining plays a crucial role in intrusion detection by enabling the identification of patterns and anomalies in network traffic that may indicate malicious activity. By leveraging data mining techniques, organizations can effectively detect, prevent, and respond to intrusions, minimizing security breaches and protecting their valuable data and systems.

Key Applications of Data Mining in Intrusion Detection:

1. **Anomaly Detection:** Data mining algorithms can analyze network traffic patterns to identify anomalous behavior that deviates from normal network activity. This can help detect intrusions that may not be detected by signature-based intrusion detection systems (IDS).
2. **Misuse Detection:** Data mining techniques can identify known attack patterns and signatures by analyzing network traffic and system logs. This can help detect intrusions that are using known vulnerabilities or exploits.
3. **Network Traffic Analysis:** Data mining algorithms can analyze network traffic data to identify suspicious patterns, such as high volumes of traffic, unusual traffic patterns, or unauthorized access attempts. This can help identify potential intrusions and prioritize further investigation.
4. **Feature Extraction:** Data mining techniques can extract relevant features from network traffic data, such as packet headers, protocol information, and payload content. These features can then be used to train intrusion detection models.
5. **Intrusion Pattern Recognition:** Data mining algorithms can identify patterns in network traffic that may indicate specific types of intrusions, such as denial-of-service attacks, port scans, or malware infections. This can help in classifying and prioritizing detected intrusions.

Common Data Mining Techniques for Intrusion Detection:

1. **Clustering:** Clustering algorithms group similar network traffic patterns together, allowing for the identification of outliers or anomalies that may indicate intrusions.
2. **Classification:** Classification algorithms can be trained on labeled network traffic data to distinguish between normal and malicious traffic.
3. **Association Rule Mining:** Association rule mining algorithms uncover relationships between network traffic events, helping to identify patterns that may indicate intrusions or malicious activities.
4. **Sequential Pattern Mining:** Sequential pattern mining algorithms identify sequences of network traffic events that may indicate intrusions or malicious activities, such as a series of failed login attempts followed by unauthorized access.
5. **Outlier Detection:** Outlier detection algorithms identify data points that deviate significantly from normal network traffic patterns, potentially signaling malicious activity.

By employing data mining techniques, organizations can effectively enhance their intrusion detection capabilities, proactively identify and respond to threats, and maintain a secure IT infrastructure.

Market basket analysis

Data mining plays a crucial role in market basket analysis, enabling retailers to uncover hidden patterns and trends in customer purchasing behavior. By analyzing large amounts of transaction data, retailers can identify frequently purchased items together, known as market baskets, and utilize these insights to develop effective marketing strategies, product placement, and promotional campaigns.

Key Applications of Data Mining in Market Basket Analysis:

1. **Market Basket Analysis:** Data mining algorithms can analyze transaction data to identify frequently purchased items together, revealing associations between products that customers tend to buy together. This information can be used to develop targeted promotions, optimize product placement, and enhance customer satisfaction.

2. **Customer Segmentation:** Data mining techniques can help segment customers into distinct groups based on their purchasing patterns, allowing retailers to tailor marketing strategies and product recommendations to specific customer segments. This personalized approach can increase customer engagement and sales.
3. **Product Recommendation:** Data mining algorithms can analyze customer purchase history and product attributes to recommend relevant products to individual customers. This personalized recommendation system can enhance customer satisfaction, increase sales, and reduce cart abandonment.
4. **Promotional Analysis:** Data mining techniques can analyze the effectiveness of promotional campaigns by evaluating sales patterns before, during, and after promotions. This analysis can help retailers optimize promotional strategies and maximize the return on investment (ROI) of marketing campaigns.
5. **Fraud Detection:** Data mining algorithms can identify unusual purchasing patterns that may indicate fraudulent activity, such as large-scale purchases with stolen credit cards or suspicious patterns of returns and exchanges. This can help retailers minimize losses due to fraud.

Common Data Mining Techniques for Market Basket Analysis:

1. **Apriori Algorithm:** The Apriori algorithm is a widely used method for identifying frequent itemsets in transaction data. It employs an iterative approach to identify itemsets with increasing support, forming the basis for market basket analysis.
2. **FP-Growth Algorithm:** The FP-Growth algorithm is another efficient method for frequent pattern mining. It constructs an FP-tree data structure to compress transaction data, enabling faster processing of large datasets.
3. **Pattern-Growth Algorithm:** The Pattern-Growth algorithm is an extension of the FP-Growth algorithm that mines frequent patterns directly from the FP-tree, further improving efficiency and scalability.
4. **Association Rule Mining:** Association rule mining algorithms identify associations between items in transaction data, revealing rules that describe the likelihood of one item being purchased given the purchase of another. These rules form the basis for market basket analysis and product recommendation systems.
5. **Clustering Algorithms:** Clustering algorithms can group customers into distinct segments based on their purchasing patterns, allowing retailers to tailor marketing strategies and product recommendations to specific customer segments.

By employing data mining techniques, retailers can gain valuable insights from their transaction data, enabling them to make informed decisions regarding product assortment, pricing strategies, promotional campaigns, and customer engagement. This data-driven approach can lead to increased sales, improved customer satisfaction, and enhanced profitability.

Banking and Finance

Data mining plays a crucial role in the banking and finance industry, enabling institutions to extract knowledge and insights from vast amounts of data to make informed decisions, improve customer service, and mitigate risks. Data mining techniques are applied across various aspects of banking and finance, including:

Credit Risk Assessment: Data mining algorithms analyze customer credit history, financial information, and other relevant data to assess creditworthiness and predict the likelihood of loan repayment. This helps banks make informed decisions about loan approvals, interest rates, and credit limits.

Fraud Detection: Data mining techniques identify patterns and anomalies in transaction data to detect fraudulent activity, such as unauthorized transactions, card misuse, or money laundering. This helps banks prevent financial losses and protect their customers.

Customer Segmentation: Data mining algorithms segment customers based on their demographics, financial behaviors, and preferences, allowing banks to tailor marketing campaigns, product offerings, and service experiences to specific customer segments. This personalized approach enhances customer engagement and satisfaction.

Customer Churn Prediction: Data mining techniques analyze customer behavior patterns to predict the likelihood of customer churn, or attrition. This helps banks identify at-risk customers and take proactive measures to retain them.

Investment Portfolio Management: Data mining algorithms analyze historical market data, financial news, and other relevant information to identify potential investment opportunities and optimize portfolio allocation. This helps financial advisors make informed investment decisions for their clients.

Regulatory Compliance: Data mining techniques help banks comply with complex regulatory requirements by identifying patterns and anomalies in financial data that may indicate potential violations. This proactive approach

reduces the risk of regulatory penalties and reputational damage.

Common Data Mining Techniques in Banking and Finance:

1. **Clustering:** Clustering algorithms group customers or transactions into distinct segments based on their characteristics, enabling data-driven segmentation strategies.
2. **Classification:** Classification algorithms predict the likelihood of a particular outcome, such as loan repayment or fraud occurrence, based on historical data and customer attributes.
3. **Association Rule Mining:** Association rule mining algorithms identify relationships between data points, uncovering patterns that may indicate customer preferences, fraud patterns, or investment opportunities.
4. **Time Series Analysis:** Time series analysis techniques analyze historical data to identify trends, seasonality, and patterns in financial markets, aiding in investment decisions and risk management.
5. **Anomaly Detection:** Anomaly detection algorithms identify data points that deviate significantly from normal patterns, potentially signaling fraudulent activity, market anomalies, or customer churn risks.

Data mining has become an indispensable tool in the banking and finance industry, enabling institutions to make data-driven decisions, enhance customer experiences, and mitigate risks. As data continues to grow exponentially, the role of data mining will become even more crucial in shaping the future of banking and finance.

▼ 20. Advanced Concepts

Introduction

Sequential Pattern Mining

Text Mining

https://www.simplilearn.com/what-is-text-mining-in-data-mining-article

Text mining is the process of transforming unstructured text into a structured format to identify meaningful patterns and new insights.

Text Mining Process

Stage	Description
Text Pre-processing	Clean and prepare the text data for analysis
Cleanup	Remove noise, correct errors, and normalize the format
Tokenization	Divide the text into individual words or tokens
Part-of-Speech (POS) Tagging	Assign grammatical categories to each token
Text Transformation (Attribute Generation)	Represent text documents as numerical vectors
Feature Selection (Attribute Selection)	Identify and select relevant features from the text data
Variable Selection	Choose a subset of important features for model building
Remove Redundant Features	Eliminate features that provide no additional information
Remove Irrelevant Features	Discard features that offer no beneficial or pertinent information
Data Mining	Apply traditional data mining techniques to the structured text data
Evaluation	Assess the performance of the data mining models

Text Mining Techniques

Technique	Description	Use Cases
Information Retrieval (IR)	Retrieving relevant documents from a large collection	Search engines, Document management systems
Natural Language Processing (NLP)	Understanding and processing human language	Machine translation, Chatbots, Text summarization

Information Extraction (IE)	Extracting structured information from unstructured text	Named entity recognition, Relationship extraction, Event extraction
Data Mining	Identifying patterns and relationships in large datasets	Customer segmentation, Fraud detection, Market analysis

Benefits of Text Mining

1. Improved decision-making
2. Increased efficiency
3. Reduced risk
4. Improved customer satisfaction
5. Enhanced innovation/Knowledge Discovery
6. Trend Analysis
7. Fraud Detection
8. Document Clustering
9. Information Retrieval

Applications

Application	Use Cases	Benefits
Customer Service	Analyzing customer feedback to identify and address pain points.	Improved customer experience, increased customer satisfaction.
Risk Management	Monitoring sentiment and extracting information from analyst reports to identify and assess potential risks.	Enhanced risk management, more informed business decisions.
Maintenance	Analyzing maintenance data to identify patterns and predict potential problems.	Reduced downtime, improved asset utilization.
Healthcare	Clustering information from medical literature to identify new research directions and potential treatment options.	Accelerated medical research, improved patient care.
Spam Filtering	Identifying and filtering spam emails to protect users from malware and phishing attacks.	Improved email security, reduced risk of cyberattacks.

Web Mining

Web mining refers to the process of discovering and extracting useful information from a large amount of data available on the World Wide Web.

Categories of Web Mining

Web Mining Type	Description	Applications
Web Content Mining	Extracts information from the content of web pages, including text, images, and videos.	Sentiment analysis, product recommendation, opinion mining
Web Structure Mining	Analyzes the links between web pages to understand the structure of the web.	Search engine optimization, identifying authoritative websites, detecting web spam
Web Usage Mining	Tracks how users interact with websites, such as their clickstream data and search queries.	Website personalization, targeted advertising, website design improvement

Process of Web Mining

Web Mining Step	Description
Data Collection	Collecting web data from various sources
Data Preprocessing	Removing irrelevant information and duplicate content
Data Integration	Transforming pre-processed data into a structured format
Pattern Discovery	Identifying patterns, trends, and relationships using web mining techniques
Evaluation	Assessing the significance and usefulness of discovered patterns

Visualization	Representing analysis results through graphs, charts, and other visualizations

Applications of Web Mining

Web Mining Application	Description
Marketing and Advertising	Analyzing consumer behavior, identifying trends, and personalizing marketing campaigns
Business Intelligence	Extracting valuable insights from web data, including competitor analysis, market trends, and customer preferences
E-commerce	Optimizing website design, personalizing product recommendations, and improving customer experience by analyzing user behavior
Fraud Detection	Detecting fraudulent activities, such as credit card fraud, identity theft, and online scams
Social Network Analysis	Understanding social dynamics, sentiment analysis, and targeted advertising by analyzing social media data

Challenges in Web Mining

Challenge	Description
Complexity of web pages	Web pages lack a unified structure and are more complex than traditional text documents.
Dynamic nature of web data	Web data is constantly changing, making it difficult to capture and analyze.
Diversity of client networks	Users have diverse interests, backgrounds, and usage patterns, making it challenging to personalize web experiences.
Data relevance	Users are often only interested in a small subset of web data, making it difficult to filter irrelevant information.
Sheer size of the web	The vast and ever-growing size of the web poses challenges for data storage and processing.

Graph Mining

Definition : *Graph Mining* is the set of tools and techniques used to

(a) analyze the properties of real-world graphs,

(b) predict how the structure and properties of a given graph might affect some application, and

(c) develop models that can generate realistic graphs that match the patterns found in real-world graphs of interest.

Process of Graph Mining

Stage	Description
Data Collection	Gathering graph data from various sources, such as social networks, biological networks, and computer networks.
Data Preprocessing	Cleaning and preparing the graph data for analysis by removing noise, correcting errors, and normalizing the format.
Feature Extraction	Identifying and extracting relevant features from the graph data, such as node attributes, edge weights, and graph-level metrics.
Model Building	Developing and training machine learning models using the extracted features to perform tasks like classification, clustering, or link prediction.
Evaluation	Assessing the performance of the trained models using appropriate metrics and refining the models as needed.

Applications

Application	Description

Frequent subgraph mining	Identifying frequently occurring patterns in graphs
Community detection	Uncovering tightly knit groups of nodes within a graph
Information diffusion and virus propagation	Modeling the spread of information or viruses through a graph
Graph kernels	Comparing graphs based on their structural similarity
Ranking on graphs	Assigning relevance scores to nodes in a graph

Spatiotemporal Mining

Spatiotemporal data mining is **the process of discovering patterns and knowledge from data that relates to both space and time**. Spatiotemporal data types allow users to describe the dynamic behavior of spatial objects over time.

Ex : location of objects over time or the movement of objects in space.

Process

Step	Description
Data Collection	Gather spatiotemporal data from various sources, such as sensors, GPS devices, and databases.
Data Preprocessing	Clean and prepare the spatiotemporal data for analysis by removing noise, correcting errors, and normalizing the data.
Feature Extraction	Identify and extract relevant features from the spatiotemporal data, such as location coordinates, time stamps, and movement patterns.
Pattern Discovery	Apply data mining algorithms to discover patterns, trends, and relationships in the extracted features. Common STDM algorithms include k-means clustering, trajectory analysis, and anomaly detection.
Pattern Evaluation	Assess the significance and usefulness of the discovered patterns using metrics like support, confidence, and lift.
Visualization	Visualize the discovered patterns using maps, charts, and other graphical representations.

A**pplications**

Application	Description
Location-Based Services	Recommending nearby restaurants, shops, and attractions based on a user's location and preferences.
Traffic Analysis	Identifying traffic congestion patterns, predicting traffic flow, and optimizing traffic signal timing.
Environmental Monitoring	Tracking the movement of pollutants, monitoring deforestation, and predicting natural disasters.
Urban Planning	Analyzing urban growth patterns, optimizing public transportation routes, and designing sustainable cities.
Public Health	Tracking disease outbreaks, identifying risk factors for disease transmission, and targeting interventions.

Trajectory Pattern Mining

Trajectory pattern mining is a data mining technique that identifies recurring sequences of movements or positions in spatial data.

It helps us to understand the movement patterns of people and objects.

Example 1: Identifying customer behavior in a shopping mall

A shopping mall can use trajectory pattern mining to understand how customers move through the mall and identify popular areas. This information can be used to improve store layout, optimize staffing levels, and target advertising campaigns. For example, the mall might discover that customers often visit the electronics store after visiting the clothing store. This information could be used to place the electronics store closer to the clothing store in order to increase sales.

Example 2: Predicting traffic congestion

A city can use trajectory pattern mining to predict traffic congestion patterns. This information can be used to improve traffic flow and reduce congestion. For example, the city might discover that traffic congestion is often caused by a certain number of cars entering a particular intersection at the same time. This information could be used to adjust traffic signal timing in order to reduce congestion.

Types of Trajectory Patterns

Types of Trajectory Patterns	Description
Sequential patterns	Represent a specific order of locations visited or traversed. For instance, a sequential pattern might be "home → grocery store → gas station → home".
Frequent patterns	Represent the most common sequences of locations or movements. They indicate the most likely paths or behaviors observed in the data.
Periodic patterns	Exhibit recurring patterns over time, such as daily, weekly, or seasonal variations in movement.
Cluster patterns	Group together trajectories that share similar characteristics, such as starting points, destinations, or overall movement patterns.
Anomaly patterns	Represent deviations from the expected or normal movement patterns, potentially indicating unusual behavior or events.

Process

Step	Description
Data Collection	Gather trajectory data from various sources.
Data Preprocessing	Clean and prepare the trajectory data for analysis.
Feature Extraction	Extract relevant features from the trajectory data.
Pattern Discovery	Apply data mining algorithms to identify trajectory patterns.
Pattern Evaluation	Assess the significance and usefulness of the discovered patterns.
Pattern Visualization	Visualize the discovered patterns using maps, charts, and other graphical representations.
Pattern Interpretation	Interpret the discovered patterns in the context of the domain and application.
Pattern Application	Utilize the discovered patterns for various applications.

Applications

Application	Description
Location-based services	Recommending places to visit, predicting user behavior, and optimizing navigation routes.
Transportation and traffic analysis	Identifying congestion patterns, optimizing traffic signal timing, and planning transportation infrastructure.
Human mobility modeling	Understanding human movement patterns, identifying migration trends, and analyzing urban dynamics.
Environmental monitoring	Tracking animal movements, monitoring pollution dispersion, and predicting natural disasters.

Security and surveillance	Detecting anomalies in movement patterns, identifying potential threats, and optimizing surveillance strategies.

Multivariate Time Series (MTS) Mining

Multivariate time series (MTS) is a collection of observations over time that has more than one time series variable. Each variable depends on its past values and other variables. This dependency can be used to forecast future values.

Process

Step	Description
Data Collection	Gather multivariate time series data from various sources, such as sensors, databases, and APIs.
Data Preprocessing	Clean and prepare the data for analysis by handling missing values, removing outliers, and normalizing the data.
Feature Extraction	Identify and extract relevant features from the time series data, such as statistical measures, time-frequency domain features, and trend features.
Model Building	Develop and train machine learning models using the extracted features to perform tasks like classification, clustering, regression, and forecasting.
Evaluation	Assess the performance of the trained models using appropriate metrics and refine the models as needed.
Visualization	Visualize the results of the analysis to gain insights into the patterns, trends, and relationships identified in the data.

A**pplications**

Field	Application
Finance	Forecasting stock prices, predicting financial crises, and detecting fraudulent activities.
Economics	Analyzing economic indicators, forecasting economic trends, and identifying economic cycles.
Engineering	Monitoring and predicting system performance, detecting faults and anomalies, and optimizing control systems.
Healthcare	Monitoring patient vital signs, predicting disease progression, and detecting adverse drug events.
Environmental Science	Analyzing climate data, predicting weather patterns, and monitoring environmental changes.

Complex data mining

21.

22.