Data Mining (DCOM) 6th Sem

2078/2079

Question Paper Solution.

Compile by Arjun Chy

Website :- www.arjun00.com.np

1. a) Define data mining with its functionalities. Mention important application of data mining

- **Data mining** is the process of discovering patterns, trends, and insights from large datasets by using statistical, mathematical, and machine learning algorithms. It involves the extraction of knowledge from data that is stored in databases, data warehouses, or other repositories.

- **Some of the key functionalities of data mining include:**

  - **Data Cleaning:** This involves the process of removing irrelevant or incomplete data, correcting inconsistencies, and resolving data conflicts.

  - **Data Integration:** This involves the process of combining data from multiple sources to create a single, consistent view of the data.

  - **Data Transformation:** This involves the process of converting raw data into a format that is more suitable for analysis.

  - **Data Reduction:** This involves the process of reducing the amount of data to be analyzed, while still maintaining the important information.

  - **Pattern Discovery:** This involves the process of identifying meaningful patterns in the data that can help in making predictions, detecting anomalies, and identifying relationships between variables.

➢ **Some important applications of data mining include:**

- **Customer Relationship Management (CRM):** Data mining is widely used in CRM to identify customer segments, predict customer behavior, and personalize marketing campaigns.

- **Fraud Detection:** Data mining is used in fraud detection to identify unusual patterns in financial transactions that may indicate fraudulent activity.

- **Healthcare:** Data mining is used in healthcare to analyze patient data, identify risk factors, and develop personalized treatment plans.

- **Retail:** Data mining is used in retail to analyze customer purchase behavior, optimize inventory management, and forecast sales.

- **Social Media:** Data mining is used in social media to analyze user behavior, identify trends, and personalize recommendations.

b) What is fact constellation schemas? Explain star schema with its advantages and disadvantages.

➢ **Fact constellation schema** is a data modeling technique used in data warehousing that involves multiple fact tables sharing common dimensions. In this schema, each fact table represents a different aspect of the business or organization, and each dimension is shared by one or more fact tables. Fact constellation schemas are also known as galaxy schemas or star schemas.

- A star schema is a specific type of fact constellation schema that is commonly used in data warehousing. In a star schema, there is a central fact table that is connected to several dimension tables. The fact table contains the measures or metrics that are being analyzed, while the dimension tables contain the attributes or characteristics that describe the measures.

- **Advantages of Star Schema**:-

- **Simplified queries:** Star schema allows for simplified and efficient queries since the fact table is linked to each dimension table through a one-to-many relationship, making it easier to join tables.

- **Fast Aggregation:** Star schema is designed to make aggregation of data much faster and efficient, as data is stored in a denormalized form which reduces the number of joins and makes query execution faster.

- **Easy to understand and maintain:** Star schema is easy to understand and maintain since it has a simple structure and allows for quick identification of data points and relationships.

- **Scalability:** Star schema is scalable, allowing for the addition of new dimensions and measures as the business needs evolve.

- **Disadvantages of Star Schema:-**

    - **Data Redundancy:** Star schema can lead to data redundancy as dimensions are often duplicated across multiple fact tables.
    - **Data Redundancy:** Star schema can lead to data redundancy as dimensions are often duplicated across multiple fact tables.

- **Limited Flexibility:** Star schema may be limited in its flexibility since it is designed for specific queries and may not be adaptable to changing business requirements or new questions.
- **Complexity in the construction phase:** Star schema may be complex to construct as it requires significant planning and design efforts to ensure the schema meets business requirements.
- **Storage requirements:** Star schema may require more storage space since it involves the duplication of dimension tables across multiple fact tables.

In summary, while star schema has several advantages, it may not be the best option for every data warehousing scenario. It is important to consider the specific business requirements, the data sources, and the analysis needs before deciding on a data modeling technique.

2. a) Describe data warehouse. Differentiate data warehouse with DBMS.

- A data warehouse is a large, centralized repository that stores data from multiple sources, in a way that facilitates analysis, reporting, and decision-making. It is designed to support business intelligence (BI) activities, such as data mining, online analytical processing (OLAP), and advanced analytics. A data warehouse is typically used by organizations to consolidate data from various transactional systems, such as customer relationship management (CRM), enterprise resource planning (ERP), and supply chain management (SCM).

- The differences between **data warehouses and database management systems (DBMS)**:-

Criteria	Data Warehouse	DBMS
Purpose	Supports decision-making and business intelligence activities	Supports transactional processing
Data Structure	Optimized for querying and reporting	Optimized for data entry and retrieval
Data Model	Follows a dimensional data model with a star or snowflake schema	Follows an entity-relationship or relational data model
Data Integration	Integrates data from multiple sources and stores it in a single repository	Stores data related to a specific application or system
Data Volume	Handles large volumes of data, typically in the terabyte or petabyte range	Handles smaller volumes of data, typically in the gigabyte or terabyte range
Data Access	Provides read-only access to data for reporting and analysis	Provides read-write access to data for transactional processing
Querying	Supports complex queries and analysis using OLAP and data mining tools	Supports simple queries and reporting using SQL
Performance	Optimized for read-intensive workloads and batch processing	Optimized for write-intensive workloads and transactional processing

b) What are the important characteristics of OLTP. Differentiate OLAP with OLTP.

- OLTP (Online Transaction Processing) is a type of database system that is optimized for managing transactional data, such as sales orders, inventory management, and financial transactions. Here are some important characteristics of OLTP systems:

  - Data is typically stored in a normalized format, which reduces data redundancy and ensures data consistency.
  - OLTP systems are optimized for read/write operations, as they need to support high volumes of transactions.
  - The focus is on maintaining data integrity and ensuring that data is accurate and up-to-date.

- OLTP systems are designed for concurrent access by multiple users, and have features such as locking and transaction isolation to prevent conflicts between transactions.

- They typically have a high degree of data integrity and security, as the data is critical to the operation of the organization.

➢ **The differences between OLAP and OLTP systems** :-

Criteria	Data Warehouse (OLAP)	Operational Database (OLTP)
Purpose	Supports complex analysis and reporting	Supports transactional processing
Data Structure	Optimized for querying and reporting, with a denormalized structure	Optimized for data entry and retrieval, with a normalized structure
Data Model	Follows a dimensional data model with a star or snowflake schema	Follows an entity-relationship or relational data model
Data Volume	Handles large volumes of historical data, typically in the terabyte or petabyte range	Handles current and recent transactional data, typically in the gigabyte or terabyte range
Querying	Supports complex queries, aggregations, and drill-downs	Supports simple queries and reporting
Performance	Optimized for read-intensive workloads and complex queries	Optimized for write-intensive workloads and transactional processing
Data Integration	Integrates data from multiple sources and stores it in a single repository	Stores data related to a specific application or system
Data Access	Provides read-only access to data for reporting and analysis	Provides read-write access to data for transactional processing
Data Integrity	May sacrifice some data integrity for performance	Requires high data integrity and accuracy
User Community	Used by business analysts, data scientists, and decision-makers	Used by operational staff, such as salespeople, customer service representatives, and inventory managers

3. a) What is clustering? Explain linear and Non-linear regression.

- **Clustering** is a technique in machine learning and data mining that involves grouping similar objects or data points together based on their characteristics or attributes. The goal of clustering is to identify natural groupings or patterns within the data, which can then be used for further analysis or decision-making.

- **Linear regression** is a statistical technique for modeling the relationship between a dependent variable and one or more independent variables. The goal of linear regression is to find the best-fit line that describes the relationship between the variables. The line is represented by the equation y = mx + b, where y is the dependent variable, x is the independent variable, m is the slope of the line, and b is the y-intercept. Linear regression assumes that the relationship between the variables is linear, meaning that the change in the dependent variable is proportional to the change in the independent variable.

- **Non-linear regression**, on the other hand, is a statistical technique for modeling the relationship between a dependent variable and one or more independent variables, where the relationship is not linear. Non-linear regression models can take many different forms, such as quadratic, exponential, or logarithmic. The goal of non-linear regression is to find the best-fit curve that describes the relationship between the variables. Non-linear regression can be used when there is no clear linear relationship between the variables.

b) Define decision tree. Explain entropy and information gain in detail.

- A decision tree is a machine learning algorithm that uses a tree-like model to make decisions or predictions based on input data. The tree consists of nodes that represent the input variables, branches that represent the possible outcomes or values of the variables, and leaf nodes that represent the final decision or prediction.

- **Entropy** is a measure of the randomness or uncertainty of a set of data. In decision tree algorithms, entropy is used to calculate the purity or homogeneity of a set of data. If a dataset is completely homogeneous, then the entropy is zero, and if the dataset is completely random, then the entropy is maximum. The formula for entropy is given by:

$$E = - \sum p(x) \log2\ p(x)$$

where E is the entropy, p(x) is the proportion of data points in the dataset that belong to class x, and log2 is the logarithm base 2.

- **Information gain** is a measure of the reduction in entropy achieved by splitting a dataset into subsets based on a particular attribute or feature. The goal of a decision tree algorithm is to select the attribute that provides the highest information gain, as this will result in the most significant reduction in entropy and the most homogeneous subsets. The formula for information gain is given by:

$$IG(S, A) = E(S) - \sum |Sv| / |S| * E(Sv)$$

where IG(S, A) is the information gain of splitting dataset S based on attribute A, E(S) is the entropy of dataset S, |Sv| is the number of data points in subset v of S, and E(Sv) is the entropy of subset v.

4. a) What are the drawbacks of K-mean algorithm? Explain Agglomerative clustering in brief.

➢ The drawbacks of the K-means algorithm include:

- **Sensitivity to initial centroids:** The K-means algorithm is sensitive to the initial centroids, which can lead to different results for different initializations. This can make it challenging to get consistent results.
- **Difficulty in handling non-convex clusters:** The K-means algorithm assumes that the clusters are convex, which may not be the case in some datasets. This can lead to suboptimal cluster assignments.
- **The need for pre-defined K:** The K-means algorithm requires the number of clusters (K) to be pre-defined, which may not always be known in advance or may be difficult to determine.

➢ Agglomerative clustering is a bottom-up hierarchical clustering algorithm that starts with each data point as its own cluster and iteratively merges the closest clusters until a single cluster containing all the data points is formed. The algorithm uses a distance metric to measure the similarity between clusters and decides which two clusters to merge based on this similarity.

➢ Agglomerative clustering has several **advantages**, including:

- **No need for pre-defined K:** Unlike K-means, agglomerative clustering does not require the number of clusters to be pre-defined, as the algorithm automatically merges clusters until a stopping criterion is met.
- **Handling non-convex clusters:** Agglomerative clustering can handle non-convex clusters, as it is not constrained by the assumption of convexity.
- **Can produce a hierarchical structure:** Agglomerative clustering can produce a hierarchical structure of clusters, which can be useful for understanding the relationships between clusters.

➢ However, agglomerative clustering also has some **disadvantages**, such as:

- **Computationally expensive:** Agglomerative clustering can be computationally expensive, especially for large datasets, as it requires calculating the distances between all pairs of data points at each iteration.
- **Sensitivity to distance metric:** The choice of distance metric can have a significant impact on the clustering results, and some metrics may be more appropriate for certain types of data than others.

b) Explain DMQL with its syntax and example.

➢ DMQL (Data Mining Query Language) is a query language designed for data mining tasks, based on the syntax of SQL (Structured Query Language). It allows users to extract knowledge from data by specifying the relevant data, the type of knowledge to be mined, the concept hierarchy, interestingness measure, pattern presentation, and visualization.

- **Syntax for DMQL query:-**

1. Syntax for the specification of task-relevant data:-

use database <database_name> or use data warehouse
<data_warehouse_name>
from <relation(s)/cube(s)> [where condition]
in relevance to <attribute_or_dimension_list>
order by <order_list>
group by <grouping_list>
having <condition>

- This syntax specifies the database or data warehouse to be used, the relation or cube to be queried, the conditions to be applied to the query, relevant attributes or dimensions, and the ordering and grouping of results.

2. Syntax for specifying the kind of knowledge to be mined:-

mine <task_type> [as <pattern_name>]
analyze {<measure(s)>}
[where <condition>]

- This syntax specifies the type of knowledge to be mined, which can include characterization, discrimination, association, classification, and prediction. It also allows for the assignment of a name to the output pattern, and the specification of measures and conditions to be applied to the output.

3. Syntax for specifying concept hierarchy:-

with hierarchy <hierarchy_name> (<dimension_name>, <parent_attribute>, <child_attribute>)

- This syntax specifies a hierarchy for a dimension, which can be used for drilling down or rolling up the data.

4. Syntax for specifying interestingness measures:-

interest <interestingness_measure> (<attribute_name>)

- This syntax specifies interestingness measures to be used for evaluating patterns, which can include support, confidence, lift, and others.

5. Syntax for pattern presentation and visualization:-

present <visualization_type> [<visualization_options>]

- This syntax specifies the presentation and visualization options for the output pattern.

- **Putting it all together, here's an example DMQL query:-**

```
use database sales_data
from sales_table
where category = 'Electronics' and date between '2022-01-01' and '2022-12-31'
in relevance to product_name, date, sales_amount
mine association rules as electronic_sales_rules
analyze {support, confidence}
where support >= 0.1 and confidence >= 0.8
with hierarchy date_hierarchy(date, month, year)
interest lift(product_name)
present scatterplot(x=product_name, y=sales_amount, color=date)
```

- This DMQL query retrieves data from the `sales_data` database, applies certain conditions and relevant attributes, mines association rules for the 'Electronics' category of products sold in 2022, analyzes those rules based on support and confidence with minimum thresholds, specifies a date hierarchy for drilling down the data, uses lift as the interestingness measure for evaluating the patterns, and presents the output as a scatterplot with product names on the x-axis, sales amounts on the y-axis, and date color-coded.

5. a) What do you mean by slice and dice, drill up and drill down in multidimensional data?

- In the field of data analysis, multidimensional data is a commonly used term, and it refers to data that can be represented in multiple dimensions. Multidimensional data can be analyzed using a process known as OLAP (Online Analytical Processing). Slice and dice, drill up, and drill down are four common operations in OLAP, and they are used to analyze multidimensional data.

- **Slice**: A slice operation selects a subset of a cube, corresponding to a single value for one or more members of the dimension. It allows the user to create a new sub-cube by selecting a single dimension from the given cube. For example, if we have a three-dimensional cube with dimensions of Time, Product, and Region, we can perform a slice operation on the Time dimension and select a particular time period. This will result in a new sub-cube that includes data only for the selected time period.

- **Dice:** A dice operation selects a sub-cube from a cube by selecting two or more dimensions. It allows the user to create a new sub-cube by selecting more than one dimension from the given cube. For example, if we have a three-dimensional cube with dimensions of Time, Product, and Region, we can perform a dice operation on Time and Region dimensions and select a particular time period and region. This will result in a new sub-cube that includes data only for the selected time period and region.

- **Drill up:** Drill up is the process of summarizing data across one or more dimensions. It allows the user to move up in the hierarchy of a dimension by aggregating data. For example, if we have a three-dimensional cube with dimensions of Time, Product, and Region, we can drill up on the Region dimension, which will result in data aggregation across all the regions.

- **Drill down:** Drill down is the process of expanding the data to a lower level of detail by adding one or more dimensions. It allows the user to move down in the hierarchy of a dimension by adding more detail to the data. For example, if we have a three-dimensional cube with dimensions of Time, Product, and Region, we can drill down on the Time dimension by adding a lower level of detail such as a specific month.

In summary, **slice** and **dice**, **drill up**, and **drill down** are important operations in OLAP, and they help users to analyze multidimensional data efficiently. These operations allow users to select, summarize, and expand data based on specific dimensions, which enables users to obtain a better understanding of the data.

b) Explain advance data mining with its important features.

- **Advanced data mining** is the use of sophisticated techniques to analyze large datasets and discover hidden patterns and relationships within them. It involves the use of machine learning algorithms, artificial intelligence, and other statistical models to identify meaningful insights that can inform decision-making processes. Here are some **important features** of **advanced data mining**:

- **Machine Learning Algorithms**: Advanced data mining uses a variety of machine learning algorithms, such as decision trees, neural networks, and support vector machines, to automatically learn patterns and relationships within data. These algorithms are designed to improve the accuracy of predictions and make it easier to identify meaningful insights.

- **Text Mining:** Advanced data mining also involves the use of text mining techniques to extract information from unstructured data such as emails, social media posts, and customer feedback. Text mining techniques can identify themes, sentiments, and patterns within text data that would be difficult to identify manually.

- **Time Series Analysis:** Time series analysis is a key feature of advanced data mining, which allows analysts to identify patterns and trends in data over time. Time series models can help predict future trends and identify anomalies in data, which can help organizations make more informed decisions.

- **Visualization:** Data visualization is an important feature of advanced data mining, which helps analysts to identify patterns and trends in data more easily. Interactive visualizations such as heat maps, scatter plots, and network diagrams can help analysts to explore data and identify hidden relationships.

- **Big Data:** Advanced data mining techniques are designed to work with big data, which refers to datasets that are too large or complex to be analyzed using traditional data mining techniques. Advanced data mining tools can analyze large volumes of data in real-time, allowing organizations to make more informed decisions.

In summary, **advanced data mining** is a powerful tool that can help organizations to identify hidden patterns and relationships within large datasets. **Machine learning algorithms, text mining, time series analysis, visualization, and big data** processing are all **important features** of advanced data mining that can help organizations to make more informed decisions.

6.Write short notes on (Any Four)

i) K-me doid algorithm

- The K-means algorithm is a clustering algorithm used in data mining and machine learning. It is an iterative algorithm that partitions the input data into K clusters based on similarity measures. The K-means algorithm is an unsupervised learning algorithm, which means that it does not require any labeled data for training.

- The algorithm works as follows:
  - Initialize K cluster centroids randomly.
  - Assign each data point to the nearest centroid.
  - Calculate the mean of each cluster and update the centroid.
  - Repeat steps 2 and 3 until convergence.

  One of the key features of the K-means algorithm is that it is computationally efficient and can handle large datasets. It is also relatively easy to implement and interpret. However, it is sensitive to the initial cluster centroids and can converge to a local minimum, rather than the global minimum. To address this, it is common practice to run the algorithm multiple times with different initializations and select the best result.

  Another important feature of the K-means algorithm is its ability to handle numeric data. It can work with continuous or discrete data and can handle missing values using imputation techniques. It is also possible to use distance metrics other than the Euclidean distance to measure similarity between data points.

  Overall, the K-means algorithm is a popular and useful tool for exploratory data analysis and clustering tasks in data mining and machine learning.

ii) Deep Learning

- **Deep Learning** is a subset of machine learning that involves the use of artificial neural networks with three or more layers. It is inspired by the structure and function of the human brain, allowing it to "learn" from large amounts of data. Deep learning technology is used in many AI applications and services, including digital assistants, credit card fraud detection, and self-driving cars.

  One of the most popular types of deep neural networks is the convolutional neural network (CNN or ConvNet). CNNs are well-suited to processing 2D data, such as images, using 2D convolutional layers.

  Deep learning algorithms are currently the most sophisticated AI architecture in use today. They include CNNs for object detection and image classification, recurrent neural networks for speech and voice recognition, long short-term memory networks for sequence prediction, generative adversarial networks for digital photo restoration and deepfake video, and deep belief networks for healthcare sectors like cancer detection.

  Deep learning outperforms traditional machine learning algorithms like Decision Trees, SVM, Naïve Bayes Classifier, and Logistic Regression. Feature extraction is only required for ML algorithms, not deep learning algorithms.

iii) Fp-growth Algorithm

- The **FP-growth algorithm** is a frequent pattern mining algorithm used in data mining and machine learning. It is an improvement over the traditional Apriori algorithm for mining frequent patterns in a dataset.

  The FP-growth algorithm generates frequent patterns without the need for candidate generation. It represents the dataset in the form of a tree called the frequent pattern tree (FP-tree). This tree structure maintains the association between the item sets, making it easier to analyze the frequent patterns.

  The algorithm works by fragmenting the database using one frequent item and creating a pattern fragment. The item sets of these fragmented patterns are then analyzed to generate frequent patterns. This process reduces the search for frequent item sets comparatively, making it more efficient and faster than the Apriori algorithm.

  The FP-growth algorithm is widely used in association rule mining, market basket analysis, and recommender systems. It is known for its ability to handle large datasets efficiently and generate meaningful insights for businesses.

iv) MOLAP

- Multidimensional On-Line Analytical Processing (**MOLAP**) is a type of OLAP that supports multidimensional views of data through array-based multidimensional storage engines. MOLAP stores data in an optimized format in a multidimensional cube, instead of in a relational database. This allows for faster querying and analysis of data, especially when the data is dense.

  MOLAP has three main components: a database server, a MOLAP server, and a front-end tool. The MOLAP server stores and manages the multidimensional data cubes, while the front-end tool provides a graphical interface for users to interact with the data.

  One of the advantages of MOLAP is that it is optimal for slice and dice operations and can perform complex calculations. However, MOLAP can be difficult to change dimensions without re-aggregation and has limitations on the amount of data it can handle. Additionally, the storage utilization may be low if the data set is sparse.

v) DMQL

- **DMQL**, or **Data Mining Query Language**, is a query language designed for supporting ad hoc and interactive data mining. It is based on the Structured Query Language (SQL) and adopts SQL-like syntax, which makes it easy to integrate with relational query languages. The language was proposed by Han, Fu, Wang, et al. for the DBMiner data mining system.

  The purpose of DMQL is to support the whole knowledge discovery process, which requires integrated systems that can deal with both patterns and data. The inductive database approach has emerged as a unifying framework for such systems, where knowledge discovery processes become querying processes that require the design of query languages.

  DMQL provides commands for specifying primitives for data mining tasks. Its syntax covers all of these primitives, including the specification of task-relevant data, the kind of knowledge to be mined, concept hierarchy specification, interestingness measure, and pattern presentation and visualization. These components are put together to form a DMQL query.

  DMQL can work with databases and data warehouses as well. It is a powerful tool for data analysts and data scientists who need to extract valuable insights from large and complex data sets.