NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE AFFILIATED TO VTU, BELAGAVI)

Fifth Semester Mid-Semester Examination BE Degree (MSE-3)
Academic year 2023 -2024
Department of Computer Science and Engineering
Artificial Intelligence and Machine Learning (21CSG53)
MSE-III SCHEME & SOLUTION

Duration: 1 Hr

Instructions
1. ***Q1 (6 Marks) & Q6 (4 Marks) - Compulsory questions.***
2. ***Q2 - Q5 Choice-based questions for 10 marks each.***
3. ***Missing Data (if any) can be suitably assumed.***

Question No		Question	Marks
1.	a.	Describe any two clustering techniques.	2
	Sol.	**1 M for each category ---------- 2 * 1 M = 2M** Clustering techniques are **K-Means Clustering** K-Means is a partitioning method that aims to divide a dataset into K clusters, where each data point belongs to the cluster with the nearest mean (centroid) **Hierarchical Clustering** Hierarchical Clustering builds a tree-like hierarchy of clusters, either in a divisive (top-down) or agglomerative (bottom-up) fashion.	**2 * 1 M = 2M**
	b.	Outline the concept of any 4 Linkage methods.	2
	Sol.	**0.5M for each Linkage methods ---------- 4 * 0.5M = 2M** **Single Linkage:** It is the Shortest Distance between the closest points of the clusters. **Complete Linkage:** It is the farthest distance between the two points of two different clusters. **Average Linkage:** It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. **Centroid Linkage:** It is the linkage method in which the distance between the centroid of the clusters is calculated.	**4 * 0.5M = 2M**
	c.	Identify the clustering algorithm that is best suited to determine customer segments based on purchasing behavior. Highlight its benefits over a partitioning-based clustering algorithm.	2
	Sol.	**Best Suited Clustering Algorithm----------0.5M** **Customer segmentation----------0.5 M** **Benefits Over Partitioning-Based Clustering Algorithm------- each 0.25 *4 M** **Best Suited Clustering Algorithm:** K-Means Clustering	**0.5M**

		Benefits Over Partitioning-Based Clustering Algorithm (e.g., K-Medoids, DBSCAN):
		• **Simplicity and Efficiency:**
		  • K-Means is computationally efficient and straightforward, making it well-suited for large datasets in customer segmentation tasks.
		• **Centroid Representation:**
		  • K-Means provides clear centroid representations for each cluster, aiding in the interpretation and characterization of customer segments.
		• **Non-Overlapping Clusters:**
		  • K-Means ensures non-overlapping clusters, simplifying the assignment of customers to distinct segments based on purchasing behavior.
		• **Global Optimization:**
		  • K-Means optimizes a global objective function, minimizing the sum of squared distances, providing a robust approach to customer segmentation.
		Customer segmentation due to its simplicity, efficiency, and the interpretability of centroid-based representations, offering advantages over some partitioning-based clustering algorithms.
			0.25*4 =1M
			
			0.5M

2	a.	Apply the K-means clustering method for the dataset given in Table 2(a) with K=2 and initial cluster centroids are object 1 and object 2. (Show till the second iteration and the centroids of the two clusters) Table 2(a): Dataset	7
	Sol.	**Initial cluster centroids are object 1 and object 2 and showing new centroid point ---------- 1 M** **Solving iteration 1…….. 3 M** **Solving Iteration 2----------3 M**	

Object	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Iteration 1:

Analysis of the dataset given below using the K-means clustering method for the following dataset with K=2 and initial cluster centroid are object 1 and object 2

object	A (x_1)	B (y_1)	Distance to (1.0, 1.0)	Distance to (1.5, 2.0)	cluster	New cluster
1	1.0	1.0	0.00	1.11	1	
2	1.5	2.0	1.11	0.00	2	
3	3.0	4.0	3.60	2.5	2	
4	5.0	7.0	7.21	6.10	2	
5	3.5	5.0	4.71	3.60	2	
6	4.5	5.0	5.31	4.24	2	
7	3.5	4.5	4.30	3.20	2	

Initial centroids:
object 1: (1.0, 1.0)
object 2: (1.5, 2.0)

$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$1 = \sqrt{(1.0-1.0)^2 + (1.0-1.0)^2} = 0$

$2 = \sqrt{(1.0-1.5)^2 + (1.0-2.0)^2} = 1.11$

$3 = \sqrt{(1.0-3.0)^2 + (1.0-4.0)^2} = 3.60$

$4 = \sqrt{(1.0-5.0)^2 + (1.0-7.0)^2} = 7.21$

$5 = \sqrt{(1.0-3.5)^2 + (1.0-5.0)^2} = 4.71$

$6 = \sqrt{(1.0-4.5)^2 + (1.0-5.0)^2} = 5.31$

$7 = \sqrt{(1.0-3.5)^2 + (1.0-4.5)^2} = 4.3$

$1 = \sqrt{(1.0-1.5)^2 + (1.0-2.0)^2} = 1.11$

$2 = \sqrt{(1.5-1.5)^2 + (2.0-2.0)^2} = 0.00$

$3 = \sqrt{(1.5-3.0)^2 + (2.0-4.0)^2} = 2.5$

$4 = \sqrt{(1.5-5.0)^2 + (2.0-7.0)^2} = 6.10$

$5 = \sqrt{(1.5-3.5)^2 + (2.0-5.0)^2} = 3.60$

$6 = \sqrt{(1.5-4.5)^2 + (2.0-5.0)^2} = 4.24$

$7 = \sqrt{(1.5-3.5)^2 + (2.0-4.5)^2} = 3.20$

Iteration 2:

object	A (x_1)	B (y_1)	Distance (1.0, 1.0)	Distance (3.5, 4.5)	cluster	New cluster
1	1.0	1.0	0.00	4.30	1	1
2	1.5	2.0	1.11	3.20	2 →	1
3	3.0	4.0	3.60	0.70	2	2
4	5.0	7.0	7.21	2.91	2	2
5	3.5	5.0	4.71	0.5	2	2
6	4.5	5.0	5.31	1.11	2	2
7	3.5	4.5	4.30	0.00	2	2

New centroid
obj 1 (1.0, 1.0)
obj 2 (3.5, 4.5)

$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$1 = \sqrt{(3.5-1.0)^2 + (4.5-1.0)^2} = 4.30$

$2 = \sqrt{(3.5-1.5)^2 + (4.5-2.0)^2} = 3.20$

$3 = \sqrt{(3.5-3.0)^2 + (4.5-4.0)^2} = 0.70$

$4 = \sqrt{(3.5-5.0)^2 + (4.5-7.0)^2} = 2.91$

$5 = \sqrt{(3.5-3.5)^2 + (4.5-5.0)^2} = 0.5$

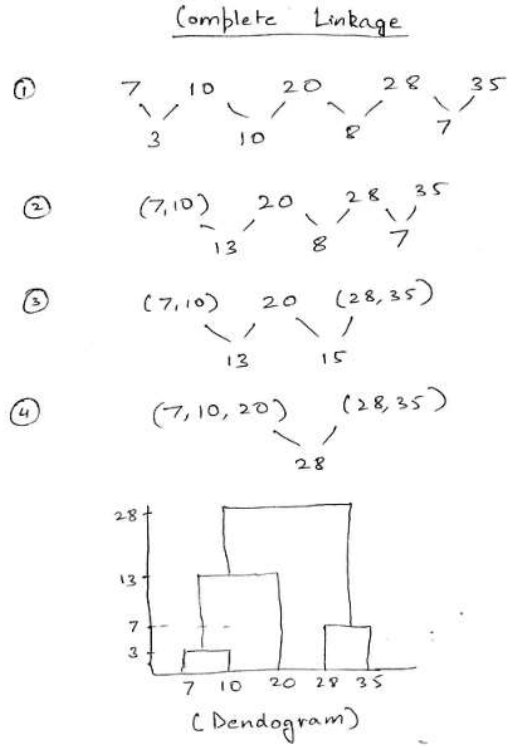
$6 = \sqrt{(3.5-4.5)^2 + (4.5-5.0)^2} = 1.11$

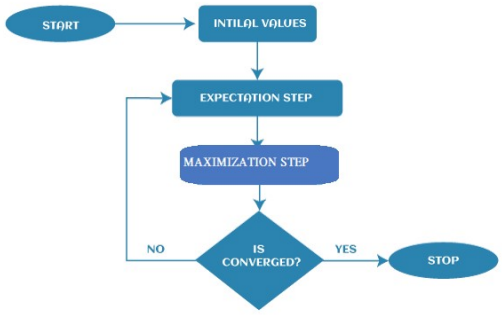
$7 = \sqrt{(3.5-3.5)^2 + (4.5-4.5)^2} = 0.00$

New centroid
obj 1: (1.25, 1.5)
obj 2: (3.9, 5.1)

3 M

3 M

	b.	Describe the basic working principles behind the functioning of Divisive Hierarchical clustering.	3
	Sol.	**Basic working principles---------- 2 marks** **Divisive Hierarchical clustering approach---------- 1 marks** Divisive Clustering is the technique that starts with all data points in a single cluster and recursively splits the clusters into smaller sub-clusters based on their dissimilarity. It is also known as, "**top-down**" clustering. It starts with all data points in a single cluster, and then recursively splits the clusters into smaller sub-clusters based on their dissimilarity. Divisive clustering is a "**divide and conquer**" approach that breaks a large cluster into smaller sub-cluster	
3	a.	Apply the Agglomerative Hierarchical Clustering using the complete linkage method for the given one-dimensional dataset of 5 points: {7, 10, 20, 28, 35},	7
	Sol.	**Complete Linkage Explanation---------- 1 M** **Solution----------5 M** **Dendogram Fig.----------2 M** 2. **Complete Linkage :** In complete link hierarchical clustering, we merge in the members of the clusters in each step, which provide the smallest maximum pairwise distance.  	**1 M** **5 M** **2 M**
	b.	Outline any two applications of the K-means clustering algorithm.	3
	Sol.	**List two Application of K – means Algo----------1 M** **Explanation of each----------2 M**	

		K-Means clustering is used in a variety of examples or business cases in real life, like: • Academic performance • Diagnostic systems • Search engines • Wireless sensor networks **Academic Performance** Based on the scores, students are categorized into grades like A, B, or C. **Diagnostic systems** The medical profession uses k-means in creating smarter medical decision support systems, especially in the treatment of liver ailments. **Search engines** Clustering forms the backbone of search engines. When a search is performed, the search results need to be grouped, and the search engines very often use clustering to do this. **Wireless sensor networks** The clustering algorithm plays the role of finding the cluster heads, which collect all the data in their respective clusters.	**1 M** **2 M**
4	a.	Show the concept of convergence within the Expectation-Maximization (EM) algorithm and explain the four key steps integral to the EM algorithm.	5
	Sol.	**EM algorithm is completed 4 steps explanation----------4M** **EM Fig4(a)----------1 M** The EM algorithm is completed mainly in 4 steps, which include I***nitialization Step, Expectation Step, Maximization Step, and convergence Step***. These steps are explained as follows:  Fig 4(a) Expectation-Maximization (EM) ○ **1st Step:** The very first step is to initialize the parameter values. Further, the system is provided with incomplete observed data with the assumption that data is obtained from a specific model. ○ **2nd Step:** This step is known as Expectation or E-Step, which is used to estimate or guess the values of the missing or incomplete data using the observed data. Further, E-step primarily updates the variables. ○ **3rd Step:** This step is known as Maximization or M-step, where we use complete data obtained from the 2nd step to update the parameter values. Further, M-step primarily updates the hypothesis.	**1 M** **4 M**

		○ **4th step:** The last step is to check if the values of latent variables are converging or not. If it gets "yes", then stop the process; else, repeat the process from step 2 until the convergence occurs.	
	b.	Apply the K-means clustering for the given one-dimensional dataset of 5 points: {7, 10, 20, 28, 35} using the Manhattan distance measure.	5
	Sol.	**Assignment of Datapoints---------- 1M** **Update Centroid values ----------1 M** **Iterations----------2 M** **Step 1: Initialization** • Choose the number of clusters (K). For illustration purposes, let's assume K = 2. • Initialize two cluster centroids arbitrarily. For simplicity, we can choose the first two points from the dataset: Centroid1 = 7 and Centroid2 = 10. **Step 2: Assignment** Assign each data point to the nearest centroid based on Manhattan distance. • Data Point 7: • Distance to Centroid1: |7 - 7| = 0 • Distance to Centroid2: |7 - 10| = 3 • Assign to Cluster 1 (Centroid1). • Data Point 10: • Distance to Centroid1: |10 - 7| = 3 • Distance to Centroid2: |10 - 10| = 0 • Assign to Cluster 2 (Centroid2). • Data Point 20: • Distance to Centroid1: |20 - 7| = 13 • Distance to Centroid2: |20 - 10| = 10 • Assign to Cluster 2 (Centroid2). • Data Point 28: • Distance to Centroid1: |28 - 7| = 21 • Distance to Centroid2: |28 - 10| = 18 • Assign to Cluster 2 (Centroid2). • Data Point 35: • Distance to Centroid1: |35 - 7| = 28 • Distance to Centroid2: |35 - 10| = 25 • Assign to Cluster 2 (Centroid2). **Step 3: Update Centroids** • Recalculate the centroids based on the mean of the data points in each cluster. • Updated Centroid1 = Mean({7}). • Updated Centroid2 = Mean({10, 20, 28, 35}). **Step 4: Iteration** • Repeat the assignment and centroid update steps until convergence. In this case, the final clustering result would be: • Cluster 1: {7} • Cluster 2: {10, 20, 28, 35}	

5	a.	Describe Principal Component Analysis Algorithm. Highlight its significance when the number of dimensions in the dataset in high.	5
	Sol.	**Algorithm Steps----------3 M** **Significance in High-Dimensional Data----------2 M** **Principal Component Analysis (PCA):** Principal Component Analysis is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving as much of the original variability as possible. It achieves this by identifying the principal components, which are linear combinations of the original features. **Algorithm Steps:** 1. **Standardization:** • Standardize the data by subtracting the mean and dividing by the standard deviation for each feature. 2. **Covariance Matrix Calculation:** • Calculate the covariance matrix for the standardized data. The covariance matrix represents the relationships between different features. 3. **Eigenvalue and Eigenvector Computation:** • Compute the eigenvalues and corresponding eigenvectors of the covariance matrix. Each eigenvector represents a principal component, and its corresponding eigenvalue indicates the amount of variance captured by that component. 4. **Sort Eigenvectors:** • Sort the eigenvectors based on their corresponding eigenvalues in descending order. The eigenvector with the highest eigenvalue represents the direction of maximum variance in the data. 5. **Projection:** • Select the top-k eigenvectors to form the transformation matrix, where k is the desired lower dimensionality. • Project the original data onto the selected eigenvectors to obtain the new, lower-dimensional representation. **Significance in High-Dimensional Data:** • **Dimensionality Reduction:** • PCA is particularly valuable when dealing with high-dimensional datasets where the number of features is large. It allows for the reduction of dimensions while retaining the most significant information. • **Variance Retention:** • PCA selects the principal components in descending order of variance, ensuring that the first few components capture the majority of variability in the data. This is crucial for maintaining meaningful information. • **Computational Efficiency:** • Reducing the number of dimensions simplifies subsequent analyses and can lead to faster model training and improved computational efficiency. • **Noise Reduction:** • By focusing on the principal components with higher variance, PCA can help reduce the impact of noise or less relevant features in the dataset. • **Visualization:**	**3 M** **2 M**

		• Lower-dimensional representations obtained through PCA are suitable for visualization purposes. They allow for a better understanding of data patterns and relationships. • **Feature Independence:** • PCA transforms the original features into a set of linearly uncorrelated variables (principal components). This can be advantageous when dealing with multicollinearity in the dataset. Principal Component Analysis is a powerful technique for dimensionality reduction, particularly in high-dimensional datasets. Its ability to capture the most significant variability in the data makes it an essential tool in various fields, including machine learning, signal processing, and data analysis.	
	b.	For the given initial set of three clusters C1 = {(1,3)(3,6)(3,5)} C2 = {(5,3),(6,7),(2,2)} C3 = {(6,5),(3,1),(2,3)} Apply the agglomerative algorithm with a single linkage cluster distance measure to produce a dendrogram tree. Note: Use Manhattan distance to compute the distance matrix.	5
	Sol.	**Manhattan distance formula----------1M** **Septs to Compute Manhattan Distance Matrix----------3 M** **Dendrogram tree---------- 1M** The agglomerative algorithm with a single linkage cluster distance measure involves iteratively merging the two closest clusters based on the Manhattan distance. Here are the steps to create the dendrogram tree: **Step 1: Compute Manhattan Distance Matrix** Calculate the Manhattan distance between each pair of clusters: $D(C_i, C_j) = \sum_{k=1}^{n} \lvert x_{ki} - x_{kj} \rvert + \lvert y_{ki} - y_{kj} \rvert$ Manhattan Distance Matrix: C1 C2 C3 C1 0 13.2 11.4 C2 13.2 0 8.4 C3 11.4 8.4 0 **Step 2: Merge Closest Clusters** Merge the closest clusters (C2 and C3) based on the minimum distance in the distance matrix. **Step 3: Update Distance Matrix** Compute the Manhattan distance between the newly formed cluster (C2-C3) and the remaining clusters. C1 C2-C3 C1 0 13.2 C2-C3 13.2 0 **Step 4: Repeat Steps 2 and 3 Until a Single Cluster is Formed**	**1 M** **3 M**

		Continue merging the closest clusters and updating the distance matrix until a single cluster is formed. Track the sequence of mergers to construct the dendrogram. (C2, C3) | C1 The dendrogram represents the hierarchical clustering process, and the height of each fusion in the tree corresponds to the Manhattan distance between the merged clusters. In this case, the dendrogram shows that C2 and C3 are first merged into a new cluster, followed by the fusion of this new cluster with C1.	**1 M**
6	a.	Examine the critical factors in evaluating and validating unsupervised learning models for anomaly detection.	4
		Examine the critical factors explanation ----------1M **Explanation for each---------- 0.25*12= 3M(based on factors mentioned)** Evaluating and validating unsupervised learning models for anomaly detection involves assessing their performance in identifying unusual patterns or outliers in data without labeled examples of anomalies. Several critical factors should be considered in this process: 1. **Ground Truth and Labeling:** • In unsupervised learning, obtaining ground truth or labeled data for anomalies is challenging. Alternative methods, such as expert domain knowledge or synthetic datasets, may be used to establish a reference for evaluating model performance. 2. **Evaluation Metrics:** • Choosing appropriate evaluation metrics is crucial. Common metrics include precision, recall, F1 score, and area under the receiver operating characteristic (ROC-AUC) curve. Precision-Recall curves are often preferred when dealing with imbalanced datasets. 3. **Imbalanced Datasets:** • Anomaly detection datasets are often highly imbalanced, with the majority of instances being normal. Care must be taken to avoid models that merely classify everything as normal. Metrics like precision and recall become essential for a comprehensive assessment. 4. **Model Interpretability:** • Understanding and interpreting the decisions made by unsupervised anomaly detection models are critical. The interpretability of results allows domain experts to make sense of identified anomalies and potentially refine the model. 5. **Robustness to Noise:** • Unsupervised learning models should be robust to noise and variations in the data. Evaluating the model's performance in the presence of noisy instances or small variations in normal patterns is essential. 6. **Scalability:** • The scalability of unsupervised anomaly detection models is vital, especially for large datasets. The model should be able to handle an increasing volume of data without sacrificing performance. 7. **Adaptability to Concept Drift:**	

		• Anomaly detection models need to adapt to changes in the data distribution over time. Evaluating their ability to handle concept drift, where the characteristics of normal and anomalous patterns evolve, is crucial. 8. **Real-Time Detection Speed:** • For applications requiring real-time anomaly detection, the speed of model predictions is a critical factor. Evaluate the latency and computational efficiency of the model in making predictions on new data. 9. **Feature Engineering:** • The choice of features used for anomaly detection can significantly impact model performance. Experiment with different feature sets and assess their relevance to improve anomaly detection accuracy. 10. **Cross-Validation Strategies:** • Employ appropriate cross-validation strategies for unsupervised learning models, such as time-based or rolling-window cross-validation, to ensure that the model generalizes well to unseen data. 11. **Domain-Specific Evaluation:** • Anomaly detection models should be evaluated in the context of specific domain requirements. The definition of what constitutes an anomaly may vary across different applications, making domain-specific evaluations essential. 12. **Explainability and Trustworthiness:** • Ensuring that the decision-making process of the model is explainable and trustworthy is crucial, particularly in applications where the consequences of false positives or false negatives are significant.	

Faculty Signature	*HOD Signature*