FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.
Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.

Machine Learning (ML-CSC701)
2023-2024
II UNIT TEST Question Bank

<table>
<tr><th>Sr. No</th><th>Questions</th></tr>
<tr><td>1</td><td>
Apply the DBSCAN algorithm with similarity threshold of 0.85 (using the similarity matrix) to the given data points and MinPts>=2 (Minimum required points in a cluster) what are core, border and noise (outliers) in the set of points given in table.
<table>
<tr><td></td><td>P1</td><td>P2</td><td>P3</td><td>P4</td><td>P5</td></tr>
<tr><td>P1</td><td>1.00</td><td>0.30</td><td>0.60</td><td>0.70</td><td>0.79</td></tr>
<tr><td>P2</td><td>0.30</td><td>1.00</td><td>0.88</td><td>0.90</td><td>0.89</td></tr>
<tr><td>P3</td><td>0.60</td><td>0.88</td><td>1.00</td><td>0.60</td><td>0.90</td></tr>
<tr><td>P4</td><td>0.70</td><td>0.90</td><td>0.60</td><td>1.00</td><td>0.80</td></tr>
<tr><td>P5</td><td>0.79</td><td>0.89</td><td>0.90</td><td>0.80</td><td>1.00</td></tr>
</table>
Sol: https://youtu.be/ZOLYaa9Jex0?si=pOpqqE6oZW7COjM_
</td></tr>
<tr><td>2

3</td><td>
The cluster has the following data point. Calculate their intra cluster distance. Also calculate its inertia .
<table>
<tr><th>X1</th><th>X2</th><th>Class</th></tr>
<tr><td>182</td><td>72</td><td>2</td></tr>
<tr><td>170</td><td>56</td><td>1</td></tr>
<tr><td>168</td><td>60</td><td>1</td></tr>
<tr><td>179</td><td>68</td><td>2</td></tr>
<tr><td>185</td><td>72</td><td>2</td></tr>
<tr><td>188</td><td>77</td><td>2</td></tr>
</table>
Apply the Xgboost on the following dataset:-
Draw the tree for only one sequence. Fill the values in the following table.
<table>
<tr><th>X1</th><th>X2</th><th>X3</th><th>Residual-1</th><th>Output of Tree_1</th><th>Residual-2</th></tr>
<tr><td>1</td><td>1</td><td>30K</td><td></td><td></td><td></td></tr>
<tr><td>1.5</td><td>1</td><td>31k</td><td></td><td></td><td></td></tr>
<tr><td>2.5</td><td>0</td><td>41k</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>0</td><td>50k</td><td></td><td></td><td></td></tr>
<tr><td>3.5</td><td>1</td><td>52k</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

4	**Plot the hyperplane for the following points:-** **(2,2),(3,2),(2,-2),(3,-2),(5,0),(6,2)(6,-2),(7,0). Also Classify the point(6,2) based on the calculated hyperplane.** **Note:- Ensure that you follow all the steps accurately, including the creation of graphs when necessary. Display the classified point by substituting values into the hyperplane equation and also depict it on the graph.**

5	**Calculate EigenValue and EigenVector for the following centered features** CentreX1 | CentreX2 1 | 0 1 | 2 3 | 4 -3 | -4 -1 | -2 -1 | 0
6 7	**Construct the RBF pattern classifier for the following dataset:-** A | B | Y 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0 **Sol:** https://youtu.be/dMGvPY9iJDc?si=PhXTlXmrFq5TOHxG; **https://youtu.be/WcUh0H51wdg?si=ZQup0km7Ci02MrbZ** **Calculate the PCA Values using the following Normalized Eigenvectors on non centered features:-** $v_1 = \begin{bmatrix} 0.59 \\ 0.81 \end{bmatrix}$ $\lambda_1 = 12.08$ $v_2 = \begin{bmatrix} -0.81 \\ 0.59 \end{bmatrix}$ $\lambda_2 = 0.32$ X1 | X2 130 | 78 128 | 80 128 | 82 126 | 78 128 | 80 128 | 82 **Sol:** **https://youtu.be/ZtS6sQUAh0c?si=CroTXqzFL4AzA1sD**

8	**Why do we need similarity or dissimilarity measure for clustering algorithms?** We need similarity or dissimilarity measures for clustering algorithms to group similar data points together and dissimilar data points apart. Clustering algorithms work by iteratively assigning data points to clusters based on their similarity to other data points. The similarity or dissimilarity measure is used to determine how similar two data points are. A good similarity or dissimilarity measure will produce results that are consistent with the desired clustering. For example, if we are clustering customers based on their purchase history, we want to use a similarity measure that takes into account the types of products they have purchased and the amounts they have spent. There are many different similarity and dissimilarity measures that can be used for clustering. Some common examples include: ● Euclidean distance ● Manhattan distance ● Cosine similarity ● Jaccard similarity ● Pearson correlation The choice of similarity or dissimilarity measure will depend on the type of data being clustered and the desired clustering outcome.
9	**Clustering algorithms are sensitive to starting points. Justify the statement with example. Also, explain impact of outliers, distance measure and noise on K-mean clustering algorithm.** Clustering algorithms are sensitive to starting points because they are often iterative in nature. This means that they start with an initial guess for the cluster assignments and then refine that guess through successive iterations. The quality of the final clustering solution will depend on the quality of the initial guess. Example: Consider the following data set of two-dimensional points: [(1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2), (4, 1), (4, 2), (5, 1), (5, 2)] We want to cluster these points into two clusters using the k-means clustering algorithm. The k-means algorithm requires us to specify the initial cluster centroids. If we choose the initial cluster centroids to be (1, 1) and (2, 2), then the final clustering solution will be as follows: Cluster 1: [(1, 1), (1, 2), (2, 1), (2, 2)] Cluster 2: [(3, 1), (3, 2), (4, 1), (4, 2), (5, 1), (5, 2)] However, if we choose the initial cluster centroids to be (3, 1) and (4, 2), then the final clustering solution will be as follows: Cluster 1: [(1, 1), (1, 2), (2, 1), (2, 2), (3, 1)] Cluster 2: [(3, 2), (4, 1), (4, 2), (5, 1), (5, 2)] As you can see, the final clustering solution is different depending on the choice of initial cluster centroids. Impact of outliers, distance measure, and noise on K-means clustering algorithm:

	● Outliers: Outliers can have a significant impact on the performance of the k-means clustering algorithm. If an outlier is chosen as an initial cluster centroid, then the algorithm may converge to a suboptimal solution. ● Distance measure: The distance measure used in the k-means clustering algorithm can also have a significant impact on the performance of the algorithm. For example, if the Euclidean distance is used, then the algorithm will be sensitive to outliers. However, if the Manhattan distance is used, then the algorithm will be more robust to outliers. ● Noise: Noise in the data can also have a negative impact on the performance of the k-means clustering algorithm. Noise can cause data points to be assigned to the wrong clusters. To mitigate the impact of outliers, distance measure, and noise on the k-means clustering algorithm, we can use the following techniques: ● Preprocessing the data: We can preprocess the data to remove outliers and noise. ● Using a robust distance measure: We can use a robust distance measure, such as the Manhattan distance, which is less sensitive to outliers. ● Using multiple initial cluster centroids: We can run the k-means clustering algorithm multiple times with different initial cluster centroids and select the clustering solution with the best quality.
10	**Why we need to use dimensionality reduction methods? PCA minimizes loss of information. Justify.** Dimensionality reduction methods are used to reduce the number of features in a dataset while preserving the most important information. This can be useful for a number of reasons, including: ● Improved performance: Machine learning algorithms often perform better on data with fewer features. This is because dimensionality reduction can help to reduce noise and overfitting. ● Reduced storage space: Datasets with fewer features require less storage space. This can be important for large datasets or datasets that need to be stored on mobile devices. ● Improved interpretability: Datasets with fewer features are easier to interpret. This can be helpful for understanding the data and identifying important patterns. PCA (Principal Component Analysis) is a dimensionality reduction method that minimizes the loss of information. PCA minimizes loss in short by projecting the data onto the principal components, which are the directions in the data space with the most variance. This means that PCA preserves the most important information in the data while discarding the less important information. In other words, PCA finds the most important dimensions in the data and then discards the less important dimensions. This is done by finding the directions in the data space where the data is most spread out. These directions are called principal components.

	The projected data is now lower-dimensional, but it still contains most of the important information from the original data. This is because the principal components capture the most variance in the data.
11	**What is bagging and boosting? How it is used to combine classifiers to improve results?** Bagging and boosting are two ensemble learning methods that can be used to combine classifiers to improve results. Bagging (bootstrap aggregating) works by creating multiple training datasets by randomly sampling with replacement from the original training dataset. Each training dataset is then used to train a separate classifier. The predictions of the individual classifiers are then averaged to produce the final prediction. Boosting works by sequentially training multiple classifiers on different weighted versions of the training dataset. The weights are adjusted after each classifier is trained to give more weight to the data points that were misclassified by the previous classifier. The final prediction is produced by taking a weighted average of the predictions of the individual classifiers. Both bagging and boosting can help to improve the performance of classifiers by reducing variance and overfitting. Bagging reduces variance by averaging the predictions of multiple classifiers. Boosting reduces overfitting by focusing on the data points that are most difficult to classify. When we use bagging or boosting, we can expect the accuracy of the classifier to improve on the test dataset. The improvement in accuracy will depend on the quality of the base classifier and the number of classifiers used in the ensemble.
12	**Explain soft margin and hard margin SVM with suitable example.** A hard margin SVM is a type of support vector machine (SVM) that tries to find a hyperplane that perfectly separates the data into two classes. A soft margin SVM is a type of SVM that allows for some misclassification of the data. Example of hard margin SVM: Suppose we have a dataset of two-dimensional data points, where each data point is labeled as either "red" or "blue". We want to use an SVM to train a classifier that can predict the label of a new data point. We can start by plotting the data points in two dimensions. Then, we can try to find a line that separates the red data points from the blue data points. If we can find a line that perfectly separates the data points, then we can use that line as a hyperplane to classify new data points. However, it is not always possible to find a line that perfectly separates the data points. In this case, we can use a soft margin SVM. Example of soft margin SVM:

	Suppose we have a dataset of two-dimensional data points, where each data point is labeled as either "red" or "blue". However, some of the data points are not perfectly separable. We can use a soft margin SVM to train a classifier that can still predict the label of a new data point, even if some of the data points are not perfectly separable. A soft margin SVM will try to find a hyperplane that maximizes the margin between the two classes, while allowing for some misclassification of the data. The margin is the distance between the hyperplane and the closest data points from each class. We can control the amount of misclassification that is allowed by using a parameter called the soft margin parameter. A higher soft margin parameter will allow for more misclassification. Soft margin SVMs are often used in real-world applications because they are more robust to noise and outliers than hard margin SVMs.
13	**What is a kernel? How you will choose appropriate kernel for problem in hand?** A kernel in machine learning is a function that takes two data points as input and returns a similarity score. Kernels are used in a variety of machine learning algorithms, including support vector machines (SVMs), kernel ridge regression, and Gaussian processes. Kernels can be used to transform data into a higher-dimensional feature space, where the data is more likely to be linearly separable. This can be useful for improving the performance of machine learning algorithms on non-linear problems. There are many different types of kernels, each with its own strengths and weaknesses. Some common kernels include: ● Linear kernel: The linear kernel simply computes the dot product of the two data points. This kernel is often used for linear problems, such as regression and classification. ● Polynomial kernel: The polynomial kernel raises the dot product of the two data points to a power. This kernel can be used for non-linear problems, but it can be computationally expensive. ● Gaussian kernel: The Gaussian kernel computes the similarity between the two data points using a Gaussian function. This kernel is often used for non-linear problems, and it is more robust to outliers than the polynomial kernel. To choose the appropriate kernel for a given problem, it is important to consider the following factors: ● Type of problem: The type of problem will determine whether a linear or non-linear kernel is needed. ● Computational complexity: Some kernels, such as the polynomial kernel, can be computationally expensive. It is important to choose a kernel that is computationally feasible for the problem at hand. ● Robustness to noise: Some kernels, such as the Gaussian kernel, are more robust to noise than others. It is important to choose a kernel that is robust to noise if the data is

	noisy.
14	Compute the principal component of following data CLASS 1 X = 2, 3, 4 Y = 1, 5, 3 CLASS 2

	X = 5, 6, 7 Y = 6, 7, 8 **Sol:** **https://youtu.be/ZtS6sQUAh0c?si=CroTXqzFL4AzA1sD**
15	Find appropriate transformation to convert non-linearly separable data to linearly separable data: Positive class: $\left\{\begin{pmatrix}2\\2\end{pmatrix},\begin{pmatrix}2\\-2\end{pmatrix},\begin{pmatrix}-2\\-2\end{pmatrix},\begin{pmatrix}-2\\2\end{pmatrix}\right.$ Negative Class: $\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix}\right\}$ Sol: https://youtu.be/03IrkMM4E6M?si=UDaO08zcr5KXSuzy