CPAN 131 Assignment3 (10%)

Please download the dataset "Experiment.csv" from BlackBoard

The dataset is related to cancer diagnosis, particularly breast cancer. The columns include various measurements and characteristics of cell nuclei present in breast cancer biopsies. Each row in the dataset represents a different biopsy sample, with the 'id' column serving as a unique identifier for each sample.

Here's a brief explanation of some of the columns:

'diagnosis': This column contains information about whether the biopsy is diagnosed as malignant (cancerous) or benign (non-cancerous).

The remaining columns contain numerical measurements of different features for each biopsy sample. These features are typically computed from images of cell nuclei, and they include mean values, standard errors, and worst (largest) values for various characteristics such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

For example:

'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean': These columns represent the mean values of these features for each cell nucleus.

'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se': These columns represent the standard errors of the corresponding features.

'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst': These columns represent the worst (largest) values of these features.

You need to apply the following steps to the dataset:

1. Load the file into a two-dimensional array. Once the loading is done, you will have an array that holds the biopsy samples.
2. Distance calculation: Now, you need to calculate the distance between each sample. Here is a process to calculate the distance between two points.

In the Euclidean plane, let point p have Cartesian coordinates (p_1, p_2) and let point q have coordinates (q_1, q_2). Then the distance between p and q is given by:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}.$$

In three dimensions, for points given by their Cartesian coordinates, the distance is

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$

In general, for points given by Cartesian coordinates in n-dimensional Euclidean space, the distance is

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}.$$

Write a class that will have a method that accepts two **array** elements and returns the distance between the arrays.

3. Write a class that will inherit the class written in Step 2. Use the method declared in the base class to calculate the distance between each sample. After the completion of the distance calculation, you will have a distance array similar to the following:

Distance array

Sample1	Sample1	0
Sample1	Sample2	Score1_2
Sample1	Sample3	Score1_3
Sample1	……	…..
Sample1	SampleN	Score1_N
Sample2	Sample1	Score2_1
Sample2	Sample2	0
Sample2	Sample3	Score2_3
Sample2	…	
Sample2	SampleN	Score2_N
So on like these for all the samples		

4. Add a method to the class declared in step 3 to calculate the TOP N closest items based on their distance for each sample. Here, N can be 3, 5,7,11, and 13. The method will work on the score array calculated in step 2 and will accept N as a method parameter. After the successful completion of this method, you will have an array similar to the following (the array is showing for N=3 and only one sample, but your code should calculate for each sample):

SamplesA	SamplesB	Score	Diagnosis
Sample1	Sample10	0.67	M
Sample1	Sample50	0.52	B
Sample1	Sample3	0.42	B

For N=3, the predicted Diagnosis for Sample 1 is "B, since B has appeared two times and M has appeared one time. Now check in the main dataset to find the value of the Diagnosis for Sample 1. If the predicted one and the actual one matches then the predicted diagnosis is correct. Complete this procedure for all the samples and calculate accuracy using the following formula:

$$\text{Accuracy (\%)} = \frac{Total\ number\ correcly\ predicted\ samples}{Total\ Number\ of\ Samples} \text{ X } 100$$

Record the accuracy for N=3.
Repeat the process for N=5, 7, 11 and 13

The assignment is completed by showing a table similar to the following:

N	**Accuracy(%)**
3	Accuracy_Score1
5	Accuracy_Score2
7	Accuracy_Score3
11	Accuracy_Score4
13	Accuracy_Score5

All the code written should follow proper object-oriented programming approaches, and the variables and class names should be done following the naming conventions.

Submission: Upload the Java files to Blackboard and demonstrate the assignment during the class. Please do not upload a ZIP file. (Without the demonstration, the assignment will not be graded)