IT 542: Pattern Recognition and Machine Learning

Assignment 6

Question 1:-

Aim: Implement Fuzzy c-means clustering algorithm. Use IRIS data to evaluate performance of the algorithm.

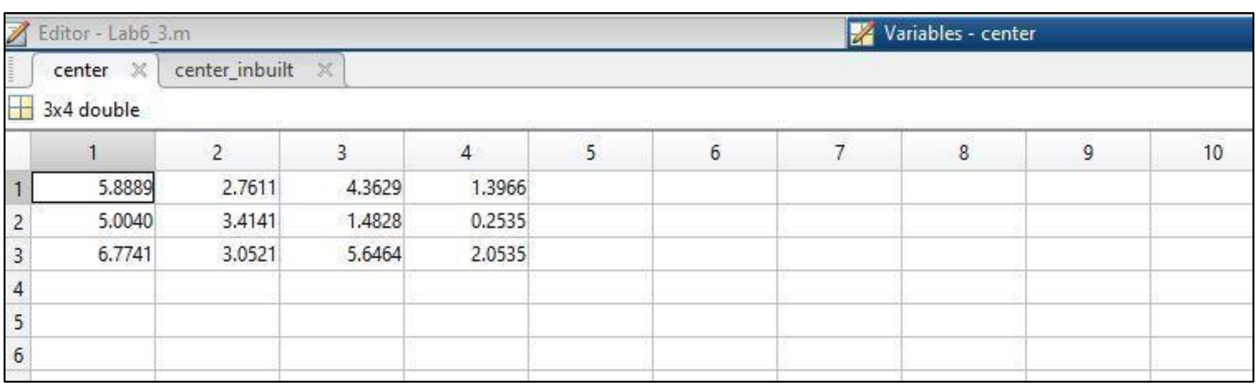
Compare your results with that of the in-built function.

Code:-

```
clc;
clear all;
load iris.dat
setosa = iris((iris(:,5)==1),:);          % data for setosa
versicolor = iris((iris(:,5)==2),:);      % data for versicolor
virginica = iris((iris(:,5)==3),:);       % data for virginica
obsv_n = size(iris, 1);                   % total number of observations

Characteristics = {'sepal length','sepal width','petal length','petal
width'};
pairs = [1 2; 1 3; 1 4; 2 3; 2 4; 3 4];
h = figure;
for j = 1:6
    x = pairs(j, 1);
    y = pairs(j, 2);
    subplot(2,3,j);
    plot([setosa(:,x) versicolor(:,x) virginica(:,x)],...
         [setosa(:,y) versicolor(:,y) virginica(:,y)], '.');
    xlabel(Characteristics{x},'FontSize',10);
    ylabel(Characteristics{y},'FontSize',10);
end

cluster_n = 3;                            % Number of clusters
expo = 2.0;                               % Exponent for U
max_iter = 100;                           % Max. iteration
min_impro = 1e-6;

% initialize fuzzy partition
U = initfcm(cluster_n, obsv_n);

if ishghandle(h)
    figure(h);
else
    for j = 1:6,
        x = pairs(j, 1);
        y = pairs(j, 2);
        subplot(2,3,j);
        plot([setosa(:,x) versicolor(:,x) virginica(:,x)],...
             [setosa(:,y) versicolor(:,y) virginica(:,y)], '.');
        xlabel(Characteristics{x},'FontSize',10);
        ylabel(Characteristics{y},'FontSize',10);
    end
end
```

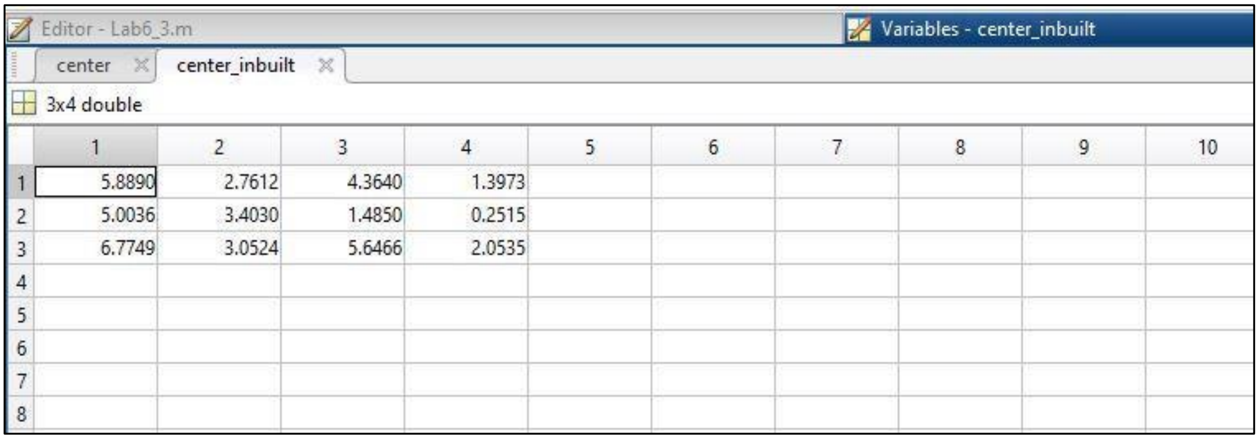
```
% iteration
for i = 1:max_iter,
    [U, center, obj] = stepfcm(iris, U, cluster_n, expo);
    fprintf('Iteration count = %d, obj. fcn = %f\n', i, obj);

    if i>1 && (abs(obj - lastobj) < min_impro)
        for j = 1:6,
            subplot(2,3,j);
            for k = 1:cluster_n,
                text(center(k, pairs(j,1)), center(k,pairs(j,2)), int2str(k),
                'FontWeight', 'bold');
            end
        end
        break;
    elseif i==1
        for j = 1:6,
            subplot(2,3,j);
            for k = 1:cluster_n,
                text(center(k, pairs(j,1)), center(k,pairs(j,2)), int2str(k),
                 'color', [0.5 0.5 0.5]);
            end
        end
    end
    lastobj = obj;
end
center=center(:,1:4);
center=center/10;

%FCM by inbuilt method
data = load('iris.csv');
data=data(:,1:4);% load some sample data
n_clusters_inbuilt = 3;                 % number of clusters
[center_inbuilt,U_inbuilt,obj_fcn_inbuilt] = fcm(data, n_clusters_inbuilt);
```

Output:-



	1	2	3	4	5	6	7	8	9	10
1	5.8889	2.7611	4.3629	1.3966						
2	5.0040	3.4141	1.4828	0.2535						
3	6.7741	3.0521	5.6464	2.0535						
4										
5										
6										

Fig 1. Centroid values without In-built Function



	1	2	3	4
1	5.8890	2.7612	4.3640	1.3973
2	5.0036	3.4030	1.4850	0.2515
3	6.7749	3.0524	5.6466	2.0535

Fig 2. Centroid values with In-built Function

Conclusion:

Centroid values computed by Fig. 1 and Fig. 2 is almost same. In program, center matrix is for centroid values without using in-built function and center_inbuilt matrix is for centroid values using In-built function.