Project 3 (SVM)

Due on 11/8/21 at 23:59 hours on Moodle
Do not submit zip file or folder. I shall NOT grade either one.
It is a group project. Submit only one report (up to three pages) for each group. Report must include sections such as problem, method or code, and results.
Project will be evaluated based on report (50%) and working (50%). You may use MATLAB or Python

In MATLAB, the binary SVM classifier is trained using 'fitcsvm'. Generate random data "X1" and "X2" as:

```
rng(1); % For reproducibility
n = 5; % Number of points per quadrant
r1 = sqrt(rand(2*n,1)); % Random radii
t1 = [pi/2*rand(n,1); (pi/2*rand(n,1)+pi)]; % Random angles for Q1 and Q3
X1 = [r1.*cos(t1) r1.*sin(t1)]; % Polar-to-Cartesian conversion
r2 = sqrt(rand(2*n,1));
t2 = [pi/2*rand(n,1)+pi/2; (pi/2*rand(n,1)-pi/2)]; % Random angles for Q2
and Q4
X2 = [r2.*cos(t2) r2.*sin(t2)];
X = [X1; X2]; % Predictors
Y = ones(4*n,1);
Y(2*n + 1:end) = -1; % Labels
```

Using this (X, Y) data, train a proper SVM classifier with fitcsvm for constraints: (i) polynomial, (ii) polynomial of degree d = 2, 3, 4, (iii) rbf, and (iv) rbf with boxcontraint 1 as shown below

```
(i) Mdl1 = fitcsvm(X,Y,'KernelFunction','polynomial','Standardize',true);
(ii) Mdl1 = fitcsvm(X,Y,'KernelFunction','polynomial', 'PolynomialOrder', 2,...
'Standardize',true);
(iii) Mdl1 = fitcsvm(X,Y,'Standardize',true,'KernelFunction','RBF',...
 %   'KernelScale','auto');
(iv) Mdl1 = fitcsvm(X,Y,'ClassNames',[false true],'Standardize',true, ...
% 'KernelFunction','rbf','BoxConstraint',1);
```

Do (1) through (4) for each case and provide ONLY results (NO CODE)
1. Plot the data:

```
figure;
gscatter(X(:,1),X(:,2),Y);
title('Scatter Diagram of Simulated Data')
```

2. Test/Validate
(a)from input data

```
XTest = [X(2,:); X(3,:); X(17,:); X(18,:)];
YTest = [Y(2,:); Y(3,:); Y(17,:); Y(18,:)];
```

```
(b)and from support vector data
XTest = Mdl1.SupportVectors
l = length(XTest)
[label,score] = predict(Mdl1,XTest)
YTest = label
table(YTest(1:l),label(1:l),score(1:l,2),'VariableNames',...
    {'TrueLabel','PredictedLabel','Score'})

3. compute and plot the score over a grid
d = 0.02; % Step size of the grid
[x1Grid,x2Grid] = meshgrid(min(X(:,1)):d:max(X(:,1)),...
min(X(:,2)):d:max(X(:,2)));
xGrid = [x1Grid(:),x2Grid(:)]; % The grid
[~,scores1] = predict(Mdl1,xGrid); % The scores
figure;
h(1:2) = gscatter(X(:,1),X(:,2),Y);
hold on
h(3) = plot(X(Mdl1.IsSupportVector,1),...
X(Mdl1.IsSupportVector,2),'ko','MarkerSize',10);
% Support vectors
contour(x1Grid,x2Grid,reshape(scores1(:,2),size(x1Grid)),[0 0],'k');
% Decision boundary
title('Scatter Diagram with the Decision Boundary')
legend({'-1','1','Support Vectors'},'Location','Best');
hold off

(4) Cross-validate the SVM classifier.
% By default, the software uses 10-fold cross-validation.
CVMdl1 = crossval(Mdl1);
misclass1 = kfoldLoss(CVMdl1);
misclass1
```