

Software Engineering and Data Science, SEIS 763: Machine Learning

Assignment #8 (100 points)

Due Date: November 15th

PRAGYA VERMA (101276204)

```
%% loading CellDNA data
CellDNA = readtable("ML_HW_Data_CellDNA.csv");
CellDNA = table2array(CellDNA);
%% Converting target dependent variable (last column) to binary values of either
0 or 1s for your two-class classification
Y = CellDNA(:,14);
Y(Y>0) = 1;
%% Assigning X variables and performing standardization
X = zscore(CellDNA(:,1:13));
%% SVM RBF Model with Box Constraints
F_score = []; Accuracy = []; Recall = []; Precision = [];
BC = [0.1, 0.5, 1, 1.5, 2, 2.5, 3];
for i = BC
    disp(['For Box Constraint = ' num2str(i)])
    SVM = fitcsvm(X,Y, 'KernelFunction', 'rbf', 'KernelScale', 1, 'BoxConstraint',i);
    [labels, scores] = predict(SVM, X);
    [ClassPerformance, OverallAccuracy] = CFM_Stats(Y, labels)
    F_score = [F_score, ClassPerformance.Fscore];
    Accuracy = [Accuracy, OverallAccuracy];
    Recall = [Recall, ClassPerformance.Fscore];
    Precision = [Precision, ClassPerformance.Fscore];
end
figure,
subplot(2,2,1), plot(BC, F_score(1, :), 'r', BC, F_score(2, :), 'b'),
title('F score'), grid on, ylim([0.3, 1.05]), legend({ 'class0', 'class1'}),
xlabel('BoxC')
subplot(2,2,2), plot(BC, Accuracy(1, :), 'r'),
title('Overall Acc'), grid on, ylim([0.3, 1.05]), legend({ 'overall'}),
xlabel('BoxC')
subplot(2,2,3), plot(BC, Recall(1, :), 'r', BC, Recall(2, :), 'b'),
title('Recall'), grid on, ylim([0.3, 1.05]), legend({ 'class0', 'class1'}),
xlabel('BoxC')
subplot(2,2,4), plot(BC, Precision(1, :), 'r', BC, Precision(2, :), 'b'),
title('Precision'), grid on, ylim([0.3, 1.05]), legend({ 'class0', 'class1'}),
xlabel('BoxC')
%% SVM RBF Model with Kernel Scales
F_score = []; Accuracy = []; Recall = []; Precision = [];
KS = [0.1, 0.5, 1, 1.5, 2, 2.5, 3];
for i = KS
    disp(['For Kernel Scale = ' num2str(i)])
    SVM = fitcsvm(X,Y, 'KernelFunction', 'rbf', 'KernelScale', i, 'BoxConstraint',1);
    [labels, scores] = predict(SVM, X);
    [ClassPerformance, OverallAccuracy] = CFM_Stats(Y, labels)

    F_score = [F_score, ClassPerformance.Fscore];
    Accuracy = [Accuracy, OverallAccuracy];
    Recall = [Recall, ClassPerformance.Fscore];
    Precision = [Precision, ClassPerformance.Fscore];
end
figure,
subplot(2,2,1), plot(KS, F_score(1, :), 'r', KS, F_score(2, :), 'b'),
```

```

title('F score'), grid on, ylim([0.6, 1.05]), legend({ 'class0', 'class1'}),
xlabel('KS')
subplot(2,2,2), plot(KS, Accuracy(1, :), 'r'),
title('Overall Acc'), grid on, ylim([0.6, 1.05]), legend({ 'overall'}),
xlabel('KS')
subplot(2,2,3), plot(KS, Recall(1, :), 'r', KS, Recall(2, :), 'b'),
title('Recall'), grid on, ylim([0.6, 1.05]), legend({ 'class0', 'class1'}),
xlabel('KS')
subplot(2,2,4), plot(KS, Precision(1, :), 'r', KS, Precision(2, :), 'b'),
title('Precision'), grid on, ylim([0.6, 1.05]), legend({ 'class0', 'class1'}),
xlabel('KS')
% ROC curve plot for class 0 by SVM RBF Kernel Scales experiment
[xpos,ypos,T,AUC0] = perfcurve(Y,scores(:,1),0);
figure, plot(xpos,ypos)
xlim([-0.05 1.05]), ylim([-0.05,1.05])
xlabel('\bf FP rate'), ylabel('\bf TP rate')
title(['\bf ROC for class 0 by SVM, AUC0 = ' num2str(AUC0)])
% ROC curve plot for class 1 by SVM RBF Kernel Scales experiment
[xpos,ypos,T,AUC1]=perfcurve(Y,scores(:,2),1);
figure, plot(xpos,ypos)
xlim([-0.05 1.05]), ylim([-0.05,1.05])
xlabel('\bf FP rate'), ylabel('\bf TP rate')
title(['\bf ROC for class 1 by SVM, AUC1 = ' num2str(AUC1)])

```