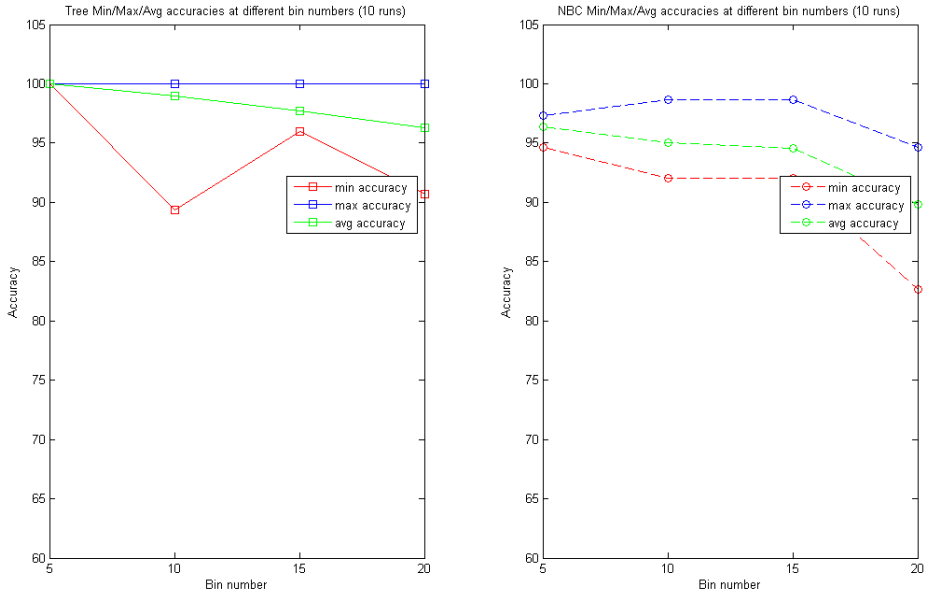
```
%#########################################################################
% Group :           Koppula Kuruba Pattan Uppari
% Students names :  Sri Chakra Goud Koppula
%                   Madan Mohan Kuruba
%                   Nishal Pattan
%                   Raj Kiran Uppari
% M# Numbers :      M12483919 ,M12484276 ,M12484225, M12483927
%#########################################################################
%#########################################################################
%##                        Machine Learning HW3                         ##
%##            Implementing ID3 and Naïve Bayes classifier              ##
%#########################################################################
%#########################################################################

%#########################################################################
%   THIS HOMEWORK USES THE FOLLOWING CLASSES
%   1) Constants        /a class that define constants values related to
%                         the data
%   2) DataHolder       /a class to hold training/testing data
%   3) DataManager      /a class to load data and make values discrete
%   4) Constants2       /a class that define constant values used to
%                         implement ID3 and NB classifier algorithms
%   5) ID3              /a class that implements the Decision tree
%                         algorithm for machine learning
%   6) node             /a class used to build nodes for the tree
%   7) NBC              /a class that implements Naive Bayes Classifier
%#########################################################################

%instantiate a new instance of the DataManager class
DM1 = DataManager();
%load the data
DM1.LoadData();

%indices for the min max and avg accuacies
MIN_ACCURACY = 1;
MAX_ACCURACY = 2;
AVG_ACCURACY = 3;

AccuraciesTree = zeros(4, 3);
AccuraciesNBC = zeros(4, 3);

%go through bin 5 10 15 20
for j=5:5:20
    %number of bins / used for discretization too
    k = j;

    % Running time for each bin
    RunningTime = 10;
    %store the accuracy of each bin, size is 10 for each run
    AccuraciesTree_EachBin = zeros(RunningTime, 1);
    AcuuraciesNBC_EachBin = zeros(RunningTime, 1);

    %run each bin 10 times

    for i=1:RunningTime


        %get a new instance of an object from class DataHolder, with data
        %already loaded into it using a function from the data manager
        %class
        DH1 = DM1.readyDataEnhanced(k);
        %insantiate a new instance from the class decision tree (ID3)
        DT1 = ID3(Constants.SETOSA, k);
```

```
        %train tree
        DT1.constructTree(DH1.Data_Training);

        %in Results first 4 columns are attributes (sepal length, sepal width, pet
        %in the 5th column it is the species of that example and the 6th column is
        %what the decision tree think that example is
        %if column 6 value is:
        %1: decision tree identify that example as setosa (target species/attribut
        %0: decision tree identify that example as not setosa
        %with higher k_values/bin there will be some examples that the tree
        %can not decide what it is, it will show as 'Not seen before'
        ResultsTree = DT1.Identify(DH1.Data_Testing);
```

Start Naive Bayes Classifier

```
        NBayesClassifier1 = NBC(k, Constants.SETOSA);

        %prepare Naive bayes classifier
        NBayesClassifier1.PrepareProbabilities(DH1.Data_Training);

        %in Results first 4 columns are attributes (sepal length, sepal width, pet
        %in the 5th column it is the species of that example and the 6th column is
        %what naive bayes classifer think that example is
        %if column 6 value is
        %1: NBC identify that example as setosa (probability of it being
        %   setosa is higher given the attributes values)
        %0: NBC identify that example as not setosa (probability of it not being
        %   setosa is higher given the attributes values)
        ResultsNBC = NBayesClassifier1.Identify(DH1.Data_Testing);

        %get indecies of correct guesses

        MatResultsTree = cell2mat(ResultsTree);
        MatResultsNBC = cell2mat(ResultsNBC);%  INSERT THE NEEDED CODE HERE

        TreeSuccessIndices = ((MatResultsTree(:, 6) == 1) & (MatResultsTree(:, 5)

        BayesSuccessIndices = ((MatResultsNBC(:, 6) == 1) & (MatResultsNBC(:, 5) =

        NumberOfSuccessTree = length(TreeSuccessIndices(TreeSuccessIndices~=0));
        NumberOfSuccessNBC = length(BayesSuccessIndices(BayesSuccessIndices~=0));%

        %calculate accuracy
        AccuracyTree = NumberOfSuccessTree/75*100;
        AccuracyNBC = NumberOfSuccessNBC/75*100;%INSERT THE NEEDED CODE HERE

        %store accuracy in vector
        AccuraciesTree_EachBin(i) = AccuracyTree;
        AcuuraciesNBC_EachBin(i) = AccuracyNBC;%INSERT THE NEEDED CODE HERE


    end

    %get min max avg for the 10 runs
    AccuraciesTree(j/5, MIN_ACCURACY) = min(AccuraciesTree_EachBin);
    AccuraciesTree(j/5, MAX_ACCURACY) = max(AccuraciesTree_EachBin);
    AccuraciesTree(j/5, AVG_ACCURACY) = mean(AccuraciesTree_EachBin);

    AccuraciesNBC(j/5, MIN_ACCURACY) = min(AcuuraciesNBC_EachBin);%INSERT THE NEED
    AccuraciesNBC(j/5, MAX_ACCURACY) = max(AcuuraciesNBC_EachBin);%INSERT THE NEED
    AccuraciesNBC(j/5, AVG_ACCURACY) = mean(AcuuraciesNBC_EachBin);%INSERT THE NEE
end
```

```
hFig = figure('Name', 'Min/Max/Avg accuracies at different bin numbers (10 runs)')
set(hFig, 'Position', [300 150 1200 800]);

%tree results
subplot(1,2,1);
TreePlotHandle = plot(5:5:20, AccuraciesTree(:, MIN_ACCURACY), '-rs', ...
                      5:5:20, AccuraciesTree(:, MAX_ACCURACY), '-bs', ...
                      5:5:20, AccuraciesTree(:, AVG_ACCURACY), '-gs');
title('Tree Min/Max/Avg accuracies at different bin numbers (10 runs)');
legend({'min accuracy', 'max accuracy', 'avg accuracy'}, ...
       'Position', [0.4, 0.65, 0, 0]);
ylim([60 105]);
ylabel('Accuracy');
xlabel('Bin number');

%NBC results
subplot(1,2,2);
NBCPlotHandle = plot(5:5:20, AccuraciesNBC(:, MIN_ACCURACY), '--ro', ...
                      5:5:20, AccuraciesNBC(:, MAX_ACCURACY), '--bo', ...
                      5:5:20, AccuraciesNBC(:, AVG_ACCURACY), '--go');
title('NBC Min/Max/Avg accuracies at different bin numbers (10 runs)');
legend({'min accuracy', 'max accuracy', 'avg accuracy'}, ...
       'Position', [0.84, 0.65, 0, 0]);
ylim([60 105]);
ylabel('Accuracy');
xlabel('Bin number');
```



Published with MATLAB® 7.10