```
%%%%%%%%%%%%%%%%%%%% GlassDecisionTree %%%%%%%%%%%%%%%%%%%%

%% Load data and variables
load('glassDataMatlab.mat');

%% Inner crossvalidation

% Rename number of inner CV-folds:
N_CV = CVin.NumTestSets;

% Pruning levels
prune = 0:10;
% Number of pruning levels tested
P = length(prune);

% minparent (tree stopping criterion)
minparent = 1;

% Impurity measures
impurity = {'gdi','deviance'};
% Number of impurity measure tested
I = length(impurity);

% Variable for classification error
Error = nan(P,I,N_CV);

for n = 1:N_CV % For each crossvalidation fold
    % print current CV-split number many times to drown out unsilenceable
    % stupid warnings
    for printmessage= 1:50
        fprintf('Crossvalidation fold %d/%d\n', n, CVin.NumTestSets);
    end

    % Extract training and validation sets
    X_train = X_par(CVin.training(n), :);
    y_train = y_par(CVin.training(n));
    X_val = X_par(CVin.test(n), :);
    y_val = y_par(CVin.test(n));

    %For each impurity measure
    for i = 1:I
        % Fit classification tree to training set
        T = classregtree(X_train, classNames(y_train), ...
            'method', 'classification', ...
            'splitcriterion', char(impurity(i)), ...
            'categorical', [], ...
            'names', attributeNames, ...
            'prune', 'on', ...
            'minparent', minparent);


        % For each pruning level
        for p = 1:P

            % Compute estimates on training and validation splits
            y_val_est = eval(T, X_val, prune(p));

            % Compute classification error
            Error(p,i,n) = sum(~strcmp(classNames(y_val),y_val_est));
        end
    end
end

%% Plot the classification error rate
mfig('Error rate');
% Sums Error over all cross-validation folds:
ValidationErrorSum = sum(Error,3)./sum(CVin.TestSize)*100;
% Plots validation error for each impurity measure:
for i = 1:I
    plot(prune,ValidationErrorSum(:,i));
    hold on;
end
title('Decision tree  - Cross-validation')
xlabel('Pruning level');
ylabel('Classification error rate (%)');
legend(impurity);
hold off;

%% Choose best pruning level P and train on all X_par

% Find lowest validation error and best pruning level and distance measure
[Evalmin, Evalmin_Index] = min(ValidationErrorSum(:))
[p0,i0] = ind2sub(size(ValidationErrorSum),Evalmin_Index)

% List of no. of classification errors for each of the CV-splits
% (To use for t-test)
E_ttest_DT = Error(p0,i0,:);
E_ttest_DT = E_ttest_DT(:)

% Train Decision tree on X_par
TBest = classregtree(X_par, classNames(y_par), ...
        'method', 'classification', ...
        'splitcriterion', char(impurity(i0)), ...
        'categorical', [], ...
        'names', attributeNames, ...
        'prune', 'on', ...
        'minparent', minparent);


% Visualise Decision tree:
view(TBest)

% Predict class
y_test_est = eval(TBest,X_test,prune(p0));
%y_test_est = str2num(cell2mat(y_test_est)); %Converts to double array

%Compute Generalization Error estimate:
E_gen_est = sum(~strcmp(classNames(y_test),y_test_est)); % Count the number of errors
E_gen_est = E_gen_est/N_test

%% Save data
save('glassDecisionTree_v4.mat');
```