

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Glass_KNN_v2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Import data and CV-partitions
load('glassDataMatlab.mat');

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

% K-nearest neighbors parameters
% Distance measures
Distance = {'euclidean', 'cityblock', 'correlation', 'cosine'};
L = length(Distance);
% Maximum number of neighbors
K = 40;

% Variable for classification error
Error = nan(K,L,N_CV);

for n = 1:N_CV % For each crossvalidation fold
    fprintf('Crossvalidation fold %d/%d\n', n, CVin.NumTestSets);

    % 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 l = 1:L % For each distance measure
        for k = 1:K % For each value of k(-nearest neighbors)

            % Use knnclassify to fit data to KNN-model with k neighbors and
            % distance measure no. l
            knn=fitcknn(X_train, y_train, 'NumNeighbors', k, 'Distance', char(Distance{
(1)));
            y_val_est=predict(knn, X_val);

            % Compute number of classification errors
            Error(k,l,n) = sum(y_val~=y_val_est); % Count the number of errors
        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 distance measure:
for l = 1:L
    plot(ValidationErrorSum(:,l));
    hold on;
end
xlabel('Number of neighbors');
ylabel('Classification error rate (%)');
legend(Distance);
hold off;

```

```
% Choose best value of K and train on all X_par

% Find lowest validation error and best K and distance measure
[Evalmin, Evalmin_Index] = min(ValidationErrorSum(:))
[k0,l0] = ind2sub(size(ValidationErrorSum),Evalmin_Index)

% List of no. of classification errors for each of the CV-splits
% (To use for t-test)
E_ttest_KNN = Error(k0,l0,:);
E_ttest_KNN = E_ttest_KNN(:)

% Train KNN with best K on X_par
knnBest=fitcknn(X_par, y_par, 'NumNeighbors', k0, 'Distance', char(Distance(l0)));

%predict
y_test_est=predict(knnBest,X_test);

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

%% Save data
save('glassKNN_v2.mat');
```