```
function [predicted_class confidence_vector probability_vector CP_accuracy error_vector] =
cluster_prediction(dataset, training_dataset, testing_dataset, training_rows, cluster_matrix, N)

num_rows = size(dataset,1);
num_testing_rows = size(testing_dataset,1);
num_errors = 0;
error_vector = zeros(num_testing_rows,1);
dataset_classes = unique(dataset(:,N+1));
num_classes = size(dataset_classes,1);

confidence_vector = [];

for i = 1 : num_testing_rows

  input_vector = testing_dataset(i,:);
  actual_class = input_vector(N+1);

  [predicted_vector predicted_class prediction_row] = find_NN(input_vector, training_dataset,N);

  x = find(cluster_matrix(training_rows(prediction_row),:) == 1); %training rows in the cluster

  class_vector = dataset(x,N+1);%the applicable cluster classes

  predicted_class = mode(class_vector);%the most frequent class

  cluster_size = size(x,2);

  cluster_entropy = entropy(class_vector);

  x = find(class_vector == predicted_class);

  num_matches = size(x,2);

  probability_vector(i) = num_matches/(cluster_size);

  confidence_vector(i) = cluster_size*log2(num_classes) - cluster_size*cluster_entropy;

  if(predicted_class != actual_class) %if true, then the allowed prediction is an error

    num_errors = num_errors + 1;
    error_vector(i) = 1;

  endif

  predicted_class_vector(i) = predicted_class;

endfor

%accuracy--------------------------------------------------------------------
CP_accuracy = 1 - num_errors/num_testing_rows;
```