```
%================================================================================
%================================================================================

%================================================================================
%VECTORIZED NORMALIZATION
%================================================================================

%================================================================================
%================================================================================

%COPYRIGHT CHARLES DAVI, 2021

%================================================================================
%LOADS THE DATASET
%================================================================================

%================================================================================
%LOADS UCI WINE DATASET
%================================================================================

%clears the memory and the command prompt
clear
clc
%--------------------------------------------------------------------------

file_path =  '/Users/charlesdavi/Desktop/Datasets/UCI/wine_full_dataset';
dataset = csvread(file_path);
num_rows = size(dataset,1);
N = size(dataset,2) - 1;

%moves the classifer to the last column----------------------------------------
temp = dataset(:,1);
dataset(:, 1) = dataset(:, N+1);
dataset(:, N+1) = temp;

%================================================================================
%NORMALIZES DATASET
%================================================================================

%calculate the average of each dimension---------------------------------------
digit_vector = log10(mean(dataset(:,1:N)))
max_dim = ceil(max(digit_vector))
min_dim = floor(min(digit_vector))

[data_categories_array final_delta] = optimize_categories_fast_N(digit_vector',1); %this is a one dimensional
categorization task
num_categories = size(data_categories_array,2);

%finds the category that contains the largest item-----------------------------
max_item = -Inf;
max_category_index = -1;

for i = 1 : num_categories

  current_category = data_categories_array{i};
  num_items = size(current_category,1);

  for j = 1 : num_items

    current_item = current_category(j);
```

```
    if(current_item > max_item)

      max_item = current_item;
      max_category_index = i;

    endif

  endfor

endfor

%finds the indexes for the largest dimensions----------------------------------
max_category = data_categories_array{max_category_index}
num_items = size(max_category,1);
counter = 1;
max_dimensions_vector = [];

for i = 1 : num_items

  current_item = max_category(i);

  for j = 1 : N

    if(digit_vector(j) == current_item)

      max_dimensions_vector(counter) = j;
      counter = counter + 1;

    endif

  endfor

endfor

%normalizes the largest category of dimensions---------------------------------
num_items = size(max_dimensions_vector,2);
weight_vector = ones(1,N+1); %used to normalize the dataset
max_accuracy = -Inf;

for i = min_dim : max_dim

  for j = 1 : num_items

    current_index = max_dimensions_vector(j);
    current_num_digits = digit_vector(current_index);

    diff = i - current_num_digits;

    weight_vector(current_index) = 10^diff;

  endfor

  weight_vector

  modified_dataset = dataset.*weight_vector;
  [accuracy output_matrix error_vector] = find_NN_dataset(modified_dataset, []);

  if(accuracy > max_accuracy)
```

```
    max_accuracy = accuracy;
    final_weight_vector = weight_vector;
    final_dataset = modified_dataset;

  endif

endfor

%==========================================================================
%GENERATE NON-MUTUALLY EXCLUSIVE CLUSTERS
%==========================================================================

tic;[cluster_matrix final_delta] = fully_vectorized_delta_clustering(final_dataset, N); toc

%==========================================================================
%TESTS ACCURACY
%==========================================================================

for i = 1 : num_rows

  x = find(cluster_matrix(i,:) == 1); %the index of the vectors in the cluster for row i

  classifier = dataset(i, N+1); %this is the classifier for the i-th vector
  cluster_classifiers = dataset(x,N+1); %these are the classifiers for the vectors in the cluster for the i-th vector

  num_errors = sum(cluster_classifiers != classifier);
  num_entries = size(x,2);
  accuracy(i) = 1 - num_errors/num_entries;

endfor

%---------------------------------

average_accuracy = mean(accuracy)*100
min_accuracy = min(accuracy)*100
max_accuracy = max(accuracy)*100

num_elements_per_cluster = sum(cluster_matrix,2);
avg_elements = mean(num_elements_per_cluster)
min_elements = min(num_elements_per_cluster)
max_elements = max(num_elements_per_cluster)

%saves an accuracy barchart

[distribution_vector] = gen_accuracy_distribution(accuracy)

x = [0:10:90]

bar(x,distribution_vector)

xlim([0 95])

file_path = '/Users/charlesdavi/Desktop/Vectorized_IP/Ionosphere_bar_chart.png';

saveas(gcf,file_path);
```