```
%========================================================================
%LOADS DATASET
%========================================================================

clear all

dataset_file = "/Users/charlesdavi/Desktop/Datasets/UCI/wine.txt";

dataset = csvread(dataset_file);
num_rows = size(dataset,1);
N = size(dataset,2) - 1;

%normalizes dataset
[weight_vector] = mean_normalization_BlackTree(dataset,N);
dataset = dataset.*weight_vector;

%finds unique class labels
class_vector = dataset(:,N+1);
class_vector = unique(class_vector);
num_classes = size(class_vector,1);

%========================================================================
%CALCULATES THE SORT RANKING
%========================================================================

for i = 1 : N

  sort_vector = sort(dataset(:,i));
  LH = sort_vector(1 : num_rows - 1);
  RH = sort_vector(2 : num_rows);
  diff_vector(i) = sum(abs(LH.-RH));

endfor

[a rank_vector] = sort(diff_vector,'descend');

%sorts dataset
sorted_dataset = sortrows(dataset,rank_vector);

%========================================================================
%RUNS PREDICTION
%========================================================================

testing_percentage = .15;
num_testing_rows = floor(num_rows*testing_percentage);

num_iterations = 100;

for i = 1 : num_iterations

  %copies classifiers and generates training / testing datasets
  testing_rows = randperm(num_rows,num_testing_rows);

  temp_dataset = sorted_dataset; %never change the sorted_dataset
  temp_dataset(:,N+2) = temp_dataset(:,N+1); %copies the classifiers
  temp_dataset(testing_rows,N+1) = -1; %flags testing rows

  [predicted_class_matrix(:,i) delta_vector prediction_vector_array modal_probability_matrix(:,i)
cluster_size_matrix(:,i)] = MASS_Sup_BlackTree(temp_dataset,N);
  [knowledge_matrix(:,i)] = calc_knowledge_supp_BlackTree(delta_vector, cluster_size_matrix(:,i), num_classes, N);
```

```
  y = find(temp_dataset(:,N+1) == -1);

  %finds testing rows
  actual_class_matrix(:,i) = temp_dataset(y,N+2);

endfor

%=============================================================================
%CALCULATES RAW ACCURACY
%=============================================================================

%finds rejections
y = find(predicted_class_matrix == -1);
temp_predicted_class_matrix = predicted_class_matrix;
temp_actual_class_matrix = actual_class_matrix;
temp_predicted_class_matrix(y) = [];
temp_actual_class_matrix(y) = [];

num_errors = sum(temp_predicted_class_matrix(:) != temp_actual_class_matrix(:));
num_predictions = size(temp_predicted_class_matrix(:),1);

raw_accuracy = 1 - num_errors/num_predictions

%=============================================================================
%CALCULATES INFORMATION CONFIDENCE
%=============================================================================

max_knowledge = max(knowledge_matrix(:));
normalized_knowledge_matrix = knowledge_matrix/max_knowledge;
confidence_matrix = min(normalized_knowledge_matrix, modal_probability_matrix);

conf_intervals = unique(confidence_matrix(:));
num_intervals = size(conf_intervals,1);

for i = 1 : num_intervals

  current_conf = conf_intervals(i);

  %finds all predictions that exceed current confidence
  x = find(confidence_matrix >= current_conf);
  inf_num_predictions(i) = size(x,1);
  num_errors = sum(predicted_class_matrix(x) != actual_class_matrix(x));
  inf_accuracy(i) = 1 - num_errors/inf_num_predictions(i);

endfor

figure, plot(inf_accuracy)

[inf_max_acc b] = max(inf_accuracy);

inf_max_acc

inf_surv_perc = (inf_num_predictions(b)/inf_num_predictions(1))*100

%=============================================================================
%CALCULATES SIZE CONFIDENCE
%=============================================================================

size_intervals = unique(cluster_size_matrix(:));
```

```
%ensures accuracy is at least the raw accuracy, removes rejections
x = find(size_intervals == 0);
size_intervals(x) = [];
num_intervals = size(size_intervals,1);

for i = 1 : num_intervals

  current_size = size_intervals(i);

  %finds all predictions that exceed current confidence
  x = find(cluster_size_matrix >= current_size);
  size_num_predictions(i) = size(x,1);
  num_errors = sum(predicted_class_matrix(x) != actual_class_matrix(x));
  size_accuracy(i) = 1 - num_errors/size_num_predictions(i);

endfor

figure, plot(size_accuracy)

[size_max_acc b] = max(size_accuracy);

size_max_acc

size_surv_perc = (size_num_predictions(b)/size_num_predictions(1))*100
```