

```

clear all

dataset_file = "/Users/charlesdavi/Desktop/Datasets/UCI/credit.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;

%calculates the sorting 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);

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

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

num_iterations = 100;

tic;
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_vector delta_vector prediction_vector_array modal_probability_vector cluster_size] =
    MASS_Sup_BlackTree(temp_dataset,N);

    %finds testing rows
    y = find(temp_dataset(:,N+1) == -1);
    actual_class_vector = temp_dataset(y,N+2);

    %finds rejections
    x = find(predicted_class_vector == -1);
    actual_class_vector(x) = [];
    predicted_class_vector(x) = [];

```

```
%calculates accuracy
num_errors = sum(actual_class_vector' != predicted_class_vector);
num_predictions = size(actual_class_vector,1);
accuracy(i) = 1 - num_errors / num_predictions;

endfor

toc

mean(accuracy)
```