```
function [delta final_rows] = supervised_delta(dataset, input_row, N)
%generates a supervised value of delta

input_vector = dataset(input_row,:);

s = std(dataset(:,1:N)); %calculates the standard deviation of the dataset in each
dimension
s = mean(s); %takes the average standard deviation
alpha = 1.5; %this is a constant used to adjust the standard deviation
s = s*alpha*N;
delta_max = s; %this is the maximum value of delta

dataset(input_row,:) = Inf;

%housekeeping variables
i = 0;
num_iterations = 25;
num_errors = 0;

while((i < num_iterations) && (num_errors == 0))

  i = i + 1;

  delta = delta_max/(num_iterations - i + 1);

  diff = sum((input_vector(1:N) .- dataset(:,1:N)).^2,2); %ignores classifiers
  matched_rows = find(diff <= delta^2);
  temp_classes = dataset(matched_rows, N+1);
  num_errors = sum(temp_classes != input_vector(N+1));

endwhile

final_rows = matched_rows;

%if true, then there was a classification error
if(num_errors != 0)

  %if true, then there was at least one error-free iteration
  if(i > 1)

    i = i - 1; %and so we back up one iteration to the previous error-free iteration
    delta = delta_max/(num_iterations - i + 1);
    final_rows = find(diff <= delta^2);

  %otherwise, the first iteration generated an error
  else
```

```
    final_rows = 0; %flag for no matches
    delta = 0;

  endif

endif

endfunction
```