```
%Copyright Charles Davi

clear all
clc

num_iterations = 250;
current_max = 0;
accuracy_matrix = [];

%calls prediction algorithm repeatedly to generate confidence curve
for i = 1 : num_iterations
  i
  tic;
  [accuracy max_cluster_size] = tempfunction();
  t(i) = toc;

  num_cols = size(accuracy_matrix,2);

  %if true, we increase the size of the accuracy matrix
  if(max_cluster_size > current_max)

    %if it's not the first iteration, we need to pad the columns
    if(i != 1)

      diff = max_cluster_size - current_max;
      accuracy_matrix(:, num_cols + 1 : num_cols + diff) = -1*ones(i-1, diff);

    endif

    accuracy_matrix(i,:) = accuracy;
    current_max = max_cluster_size;

  %if true, we pad the accuracy vector
  elseif(max_cluster_size < current_max)


    diff = current_max - max_cluster_size;
    temp = -1*ones(1, current_max + 1);
    temp(1 : max_cluster_size + 1) = accuracy;

    accuracy_matrix(i,:) = temp;

  %otherwise, they're the same size
  else

    accuracy_matrix(i,:) = accuracy;
```

```
  endif

endfor

%calculates accuracy
for i = 1 : current_max

  temp_accuracy = accuracy_matrix(:,i);

  x = find(temp_accuracy == -1);
  temp_accuracy(x) = []; %deletes null entries

  x = isnan(temp_accuracy);
  temp_accuracy(x) = []; %deletes NaN entries

  final_accuracy_vector(i) = mean(temp_accuracy);

endfor

figure, plot(final_accuracy_vector)

raw_accuracy = final_accuracy_vector(1)
max_accuracy = max(final_accuracy_vector)

average_runtime = mean(t)
```