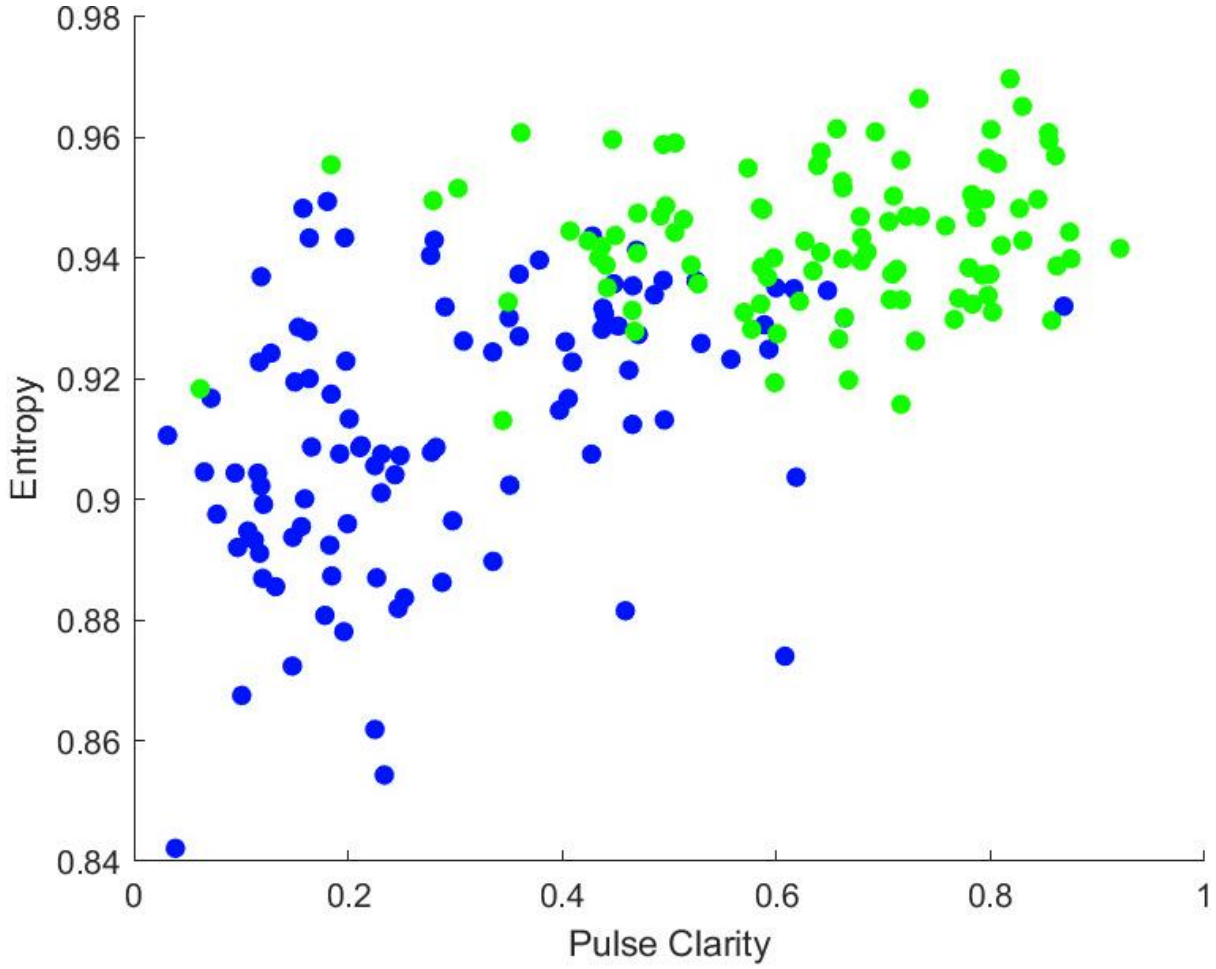
MIR ASSIGNMENT - 2

SubTask-1

Blues (in Green) v/s Disco (in Blue)
MATLAB Script:

```
scatter(pulseclarity_arr_x, entropy_arr_x, 'green', 'filled')
hold on
scatter(pulseclarity_arr_y, entropy_arr_y, 'blue', 'filled')
xlabel('Pulse Clarity')
ylabel('Entropy')
title(graphtitle)
drawnow
exportgraphics(gca, figurename)
```

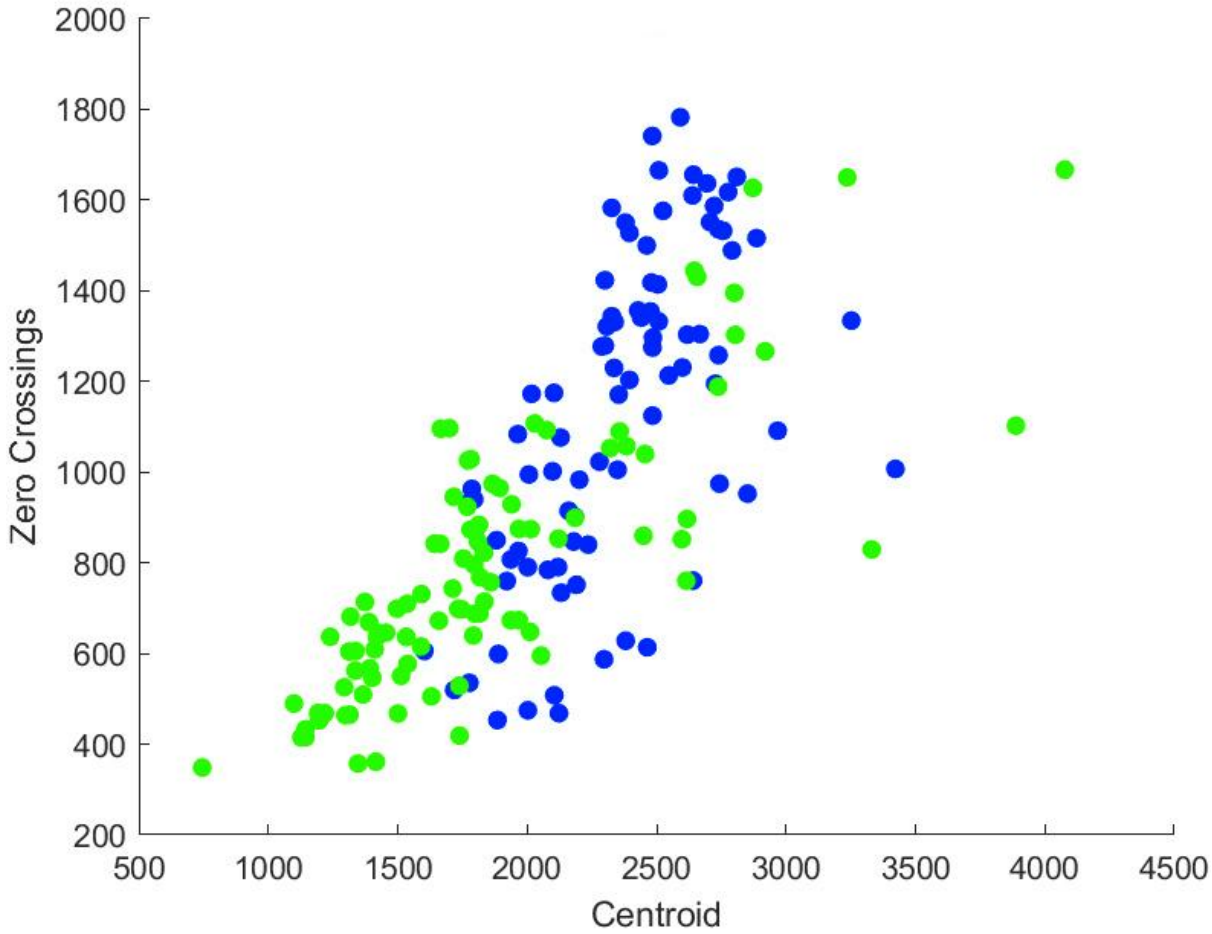
Scatterplot:



Jazz (in Green) v/s Rock (in Blue)
MATLAB Script:

```
scatter(centroid_arr_x, zero_crossing_arr_x, 'green', 'filled')
hold on
scatter(centroid_arr_x, zero_crossing_arr_y, 'blue', 'filled')
xlabel('Centroid')
ylabel('Zero Crossings')
title(graphtitle)
drawnow
exportgraphics(gca, figurename)
```

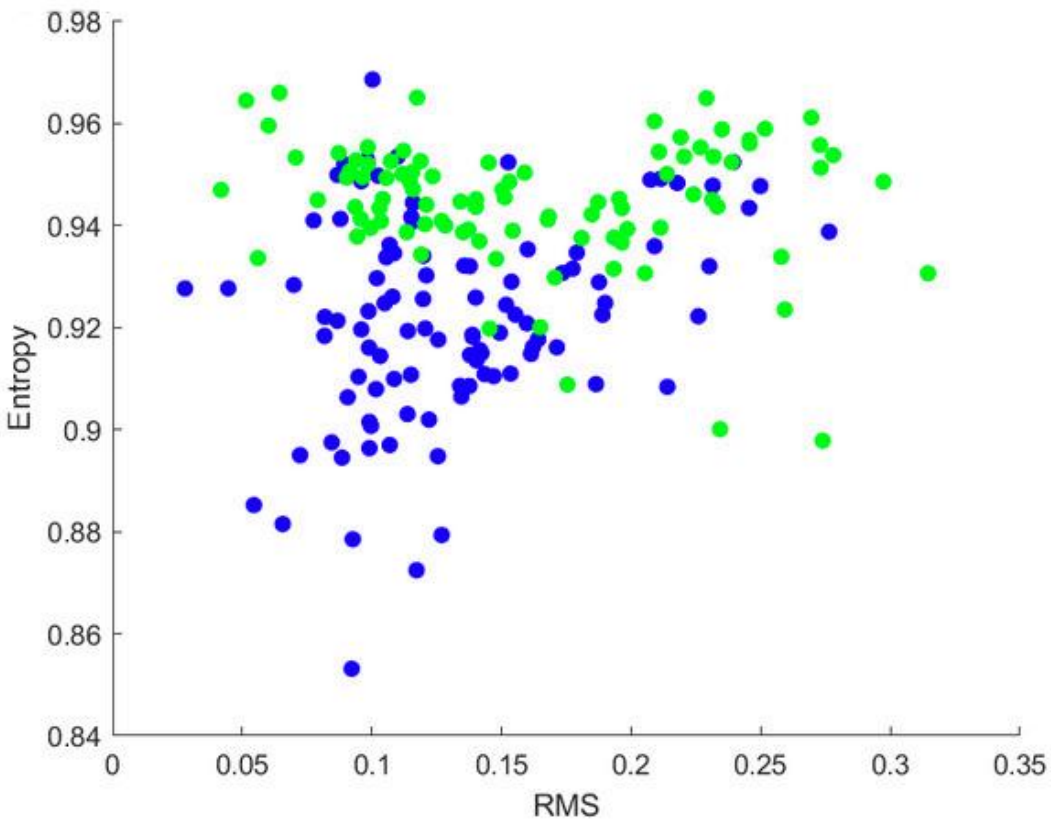
Scatterplot:



Metal (in Green) v/s Country (in Blue)
MATLAB Script:

```
scatter(rms_arr_x, entropy_arr_x, 'green', 'filled')
hold on
scatter(rms_arr_x, entropy_arr_y, 'blue', 'filled')
xlabel('RMS')
ylabel('Entropy')
title(graphtitle)
drawnow
exportgraphics(gca, figurename)
```

Scatterplot:

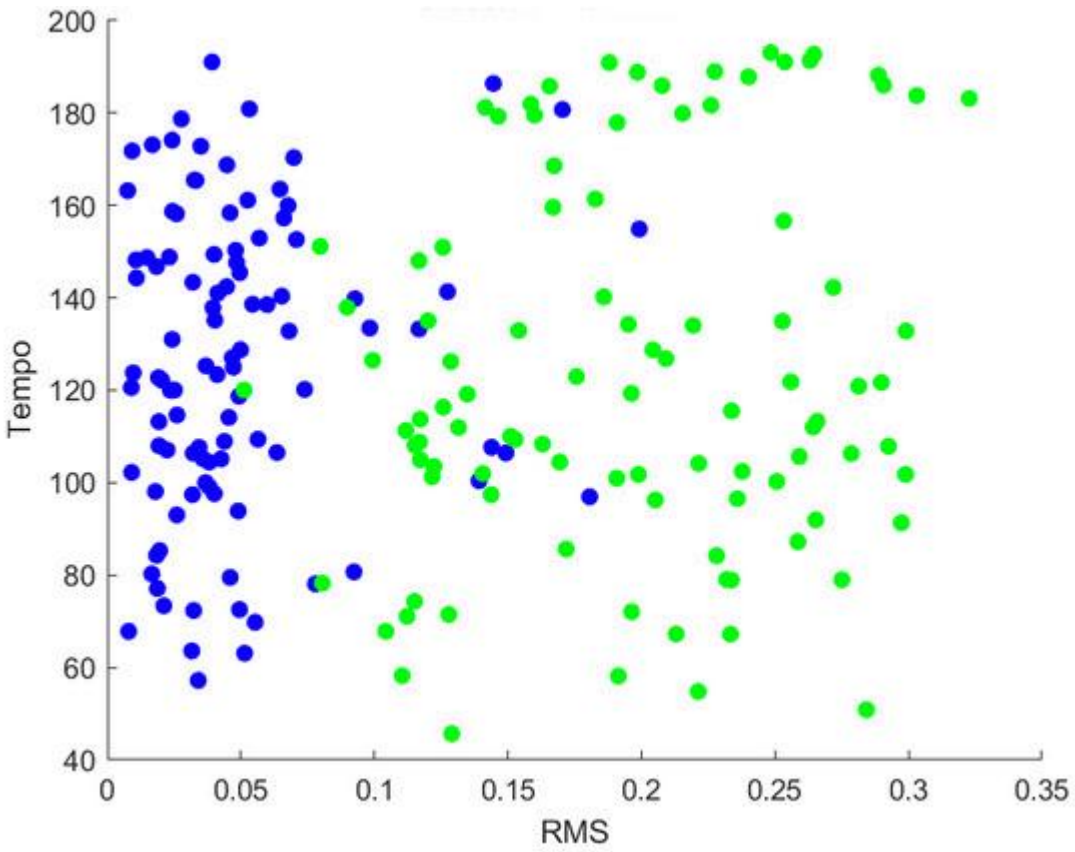


Hiphop (in Green) v/s Classical (in Blue)

MATLAB Script:

```
scatter(rms_arr_x, tempo_arr_x, 'green', 'filled')
hold on
scatter(rms_arr_x, tempo_arr_y, 'blue', 'filled')
xlabel('RMS')
ylabel(Tempo)
title(graphtitle)
drawnow
exportgraphics(gca, figurename)
```

Scatterplot:



SubTask-2

The similarity matrix of the audio files were obtained using the `mirsimatrix` function of the MIRtoolbox and saved on the local machine using the `exportgraphics` command.
The script of generating and storing the similarity matrices is given below and was used to obtain the similarity matrices for each folder individually.
For Genre Classification of these saved similarity matrices, convolution neural network(CNN) was used using tensorflow and KERAS.

```
for k = 0:9
    name = sprintf('rock.0000%d.wav', k)
    a = mirgetdata(mirspectrum(miraudio(name),'Frame',0.5)); %Get the spectrogram assuming you are using
a frame length of 500ms (which you must change by the way) and default overlap
    time_frames=size(a,2); %this tells you the number of frames
    for m=1:time_frames,
    for n=1:time_frames,
    Vi=a(:,m);
    Vj=a(:,n);
    D(m,n)=sum(Vi.*Vj)./(sqrt(sum(Vi.^2))*sqrt(sum(Vj.^2)));
    end
    end
    imagesc(D) %this will allow you to view your similarity matrix
    title('')
    xlabel('')
    ylabel('')
    xticks('')
    yticks('')
    save_song = sprintf('rock%d.png', k)
    exportgraphics(gca,save_song)

end

for k = 10:99
    name = sprintf('rock.000%d.wav', k)
    a = mirgetdata(mirspectrum(miraudio(name),'Frame',0.5)); %Get the spectrogram assuming you are using
a frame length of 500ms (which you must change by the way) and default overlap
    time_frames=size(a,2); %this tells you the number of frames
    for m=1:time_frames,
    for n=1:time_frames,
    Vi=a(:,m);
    Vj=a(:,n);
    D(m,n)=sum(Vi.*Vj)./(sqrt(sum(Vi.^2))*sqrt(sum(Vj.^2)));
    end
    end
    imagesc(D) %this will allow you to view your similarity matrix
    title('')
    xlabel('')
    ylabel('')
    xticks('')
    yticks('')
    save_song = sprintf('rock%d.png', k)
    exportgraphics(gca,save_song)

end
```