
ECE 196 Predicting Musical Genres Using Machine Learning

Table of Contents

Importing Model Data	1
Plotting	3
Importing TEST Data	10
Plotting	11
KNN Model	18

Daniel Lopez Villa -- Jimmy Do A14870615 -- A15110598 An Exploratory Study of the k-Nearest Neighbors Algorithm

Importing Model Data

```
clear; close all; clc;
audioFeatures = readtable('audioFeatures.csv');
trackInfo = readtable('trackInfo.csv');

trackInfo(:,1) = []; %delete track number column
genreTags = table2cell(trackInfo(:,3)); %extract genreTag data
genreList = ["Classical" , "80s" , "Hip Hop" , "Metal"];

numSongs = height(audioFeatures);
songs = 1:numSongs; %vector of 1:song numbers
numGenres = 4;

%genre list default to 80s
Genre = string(80*ones(numSongs,1))+"s";

songMean = zeros(numGenres,1);
average = zeros(numGenres,1);

%organizing labels/data
pattern = lettersPattern;
audioFeaturesLabels =
    extract(erase(string(table2cell(audioFeatures(1,1:2*width(audioFeatures)))),"_"),

%remove unwanted features
audioFeaturesLabels(:,end) = [];
audioFeaturesLabels(:,12:16) = [];

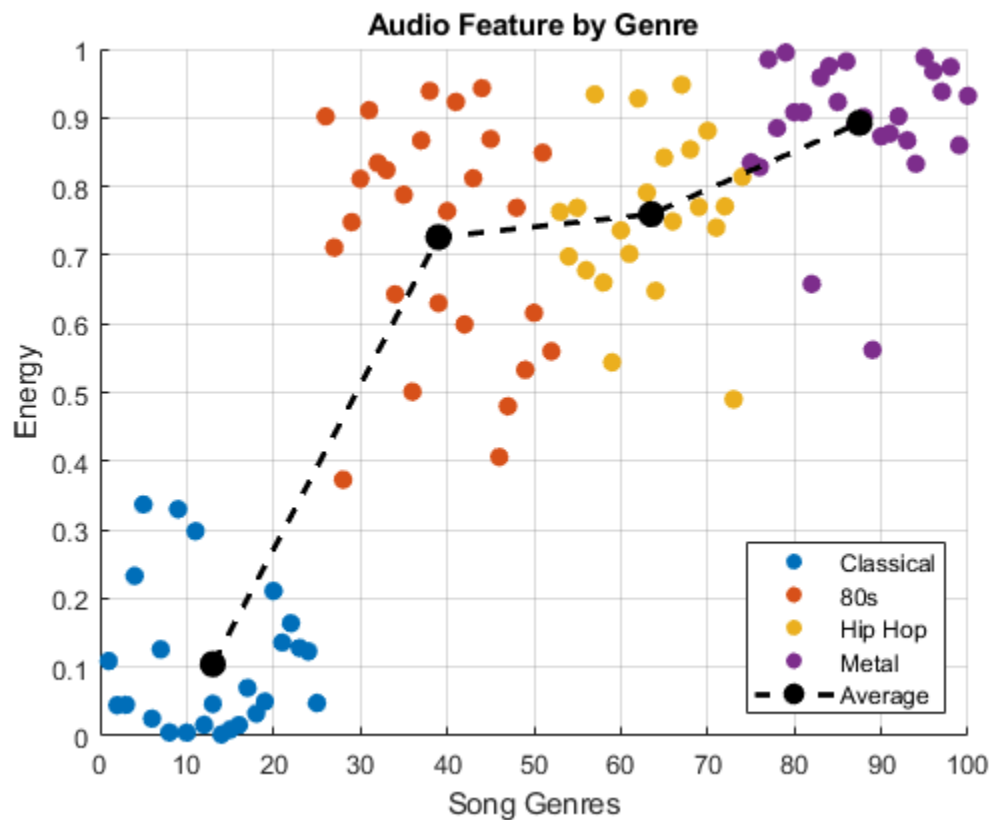
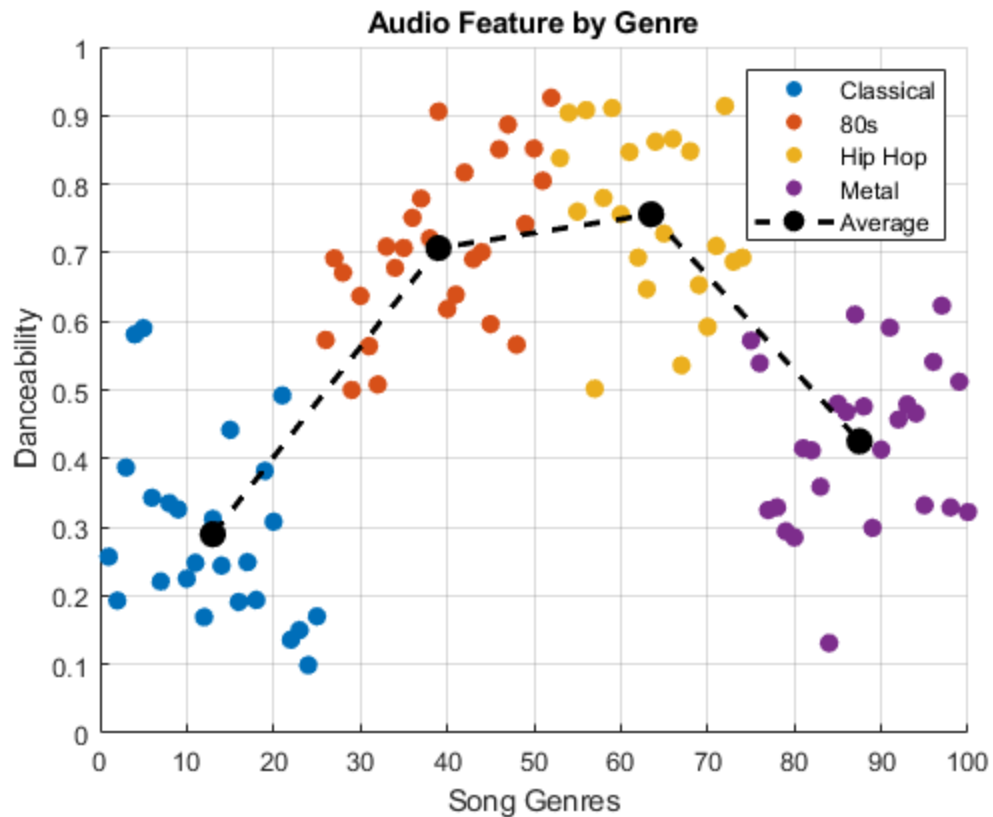
%remove unwanted features
audioFeaturesData = audioFeatures(:,2:2*width(audioFeatures));
audioFeaturesData(:,end) = [];
audioFeaturesData(:,12:16) = [];
```

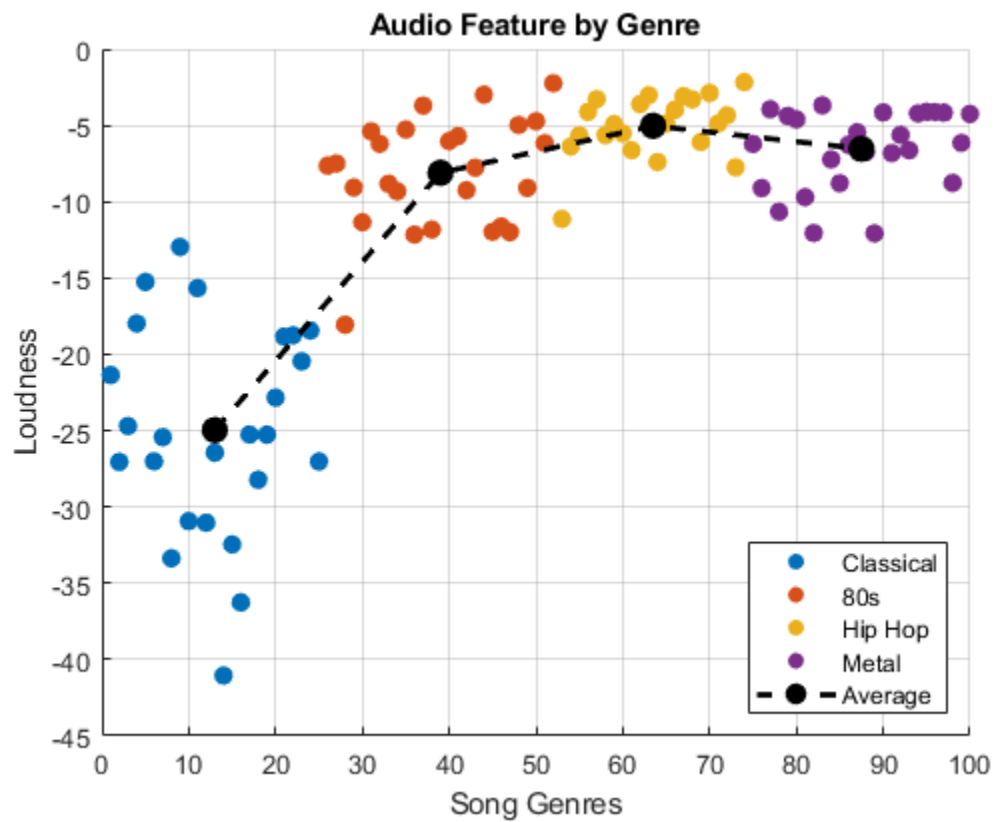
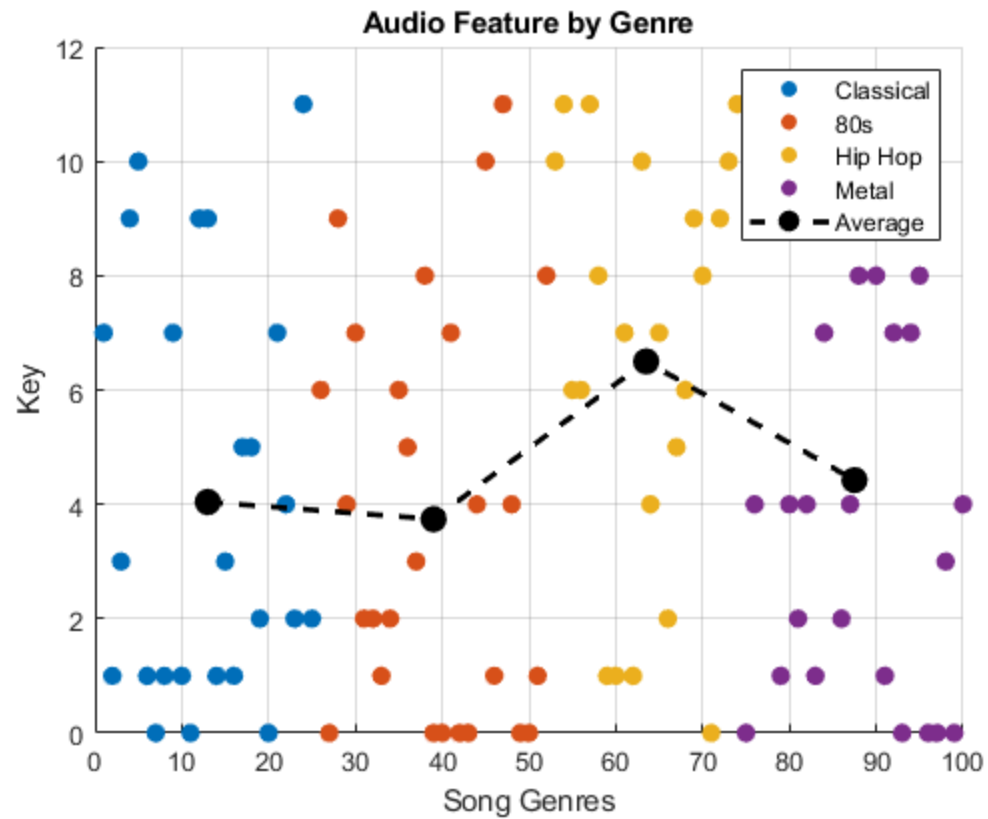
```
audioFeaturesDataNew =  
    zeros(height(audioFeatures),width(audioFeaturesData));  
for i = 1:height(audioFeatures)  
    for k=1:numGenres  
        if (contains(genreTags{i},lower(genreList(k)))) == 1  
            Genre(i,1) = genreList(k); end  
        end  
        for j = 1 : width(audioFeaturesData)  
            audioFeaturesDataNew(i,j) =  
                str2num(cell2mat(table2array(audioFeaturesData(i,j))));  
            if (i==1) audioFeaturesLabels{j}(1) =  
                upper(audioFeaturesLabels{j}(1));  
            end  
        end  
    end  
end  
  
songsPerGenre = zeros(numGenres+1,1);  
for i = 1:numGenres  
    for j = 1:width(audioFeaturesDataNew)  
        audioFeat4{i}(:,j) = audioFeaturesDataNew(find(Genre ==  
            genreList(i)),j);  
        end  
        songsPerGenre(i+1) = height(audioFeat4{i});  
    end  
end  
  
%present final data set  
Dancibility = audioFeaturesDataNew(:,1);  
Energy = audioFeaturesDataNew(:,2);  
Key = audioFeaturesDataNew(:,3);  
Loudness = audioFeaturesDataNew(:,4);  
mode = audioFeaturesDataNew(:,5);  
Speechiness = audioFeaturesDataNew(:,6);  
Acousticness = audioFeaturesDataNew(:,7);  
Instrumentalness = audioFeaturesDataNew(:,8);  
Liveness = audioFeaturesDataNew(:,9);  
Valence = audioFeaturesDataNew(:,10);  
Tempo = audioFeaturesDataNew(:,11);  
Duration = audioFeaturesDataNew(:,12);  
  
musicDataSet =  
    table(Dancibility,Energy,Valence,Acousticness,Instrumentalness,Genre);  
LabelsUsed =  
    ["Dancibility","Energy","Valence","Acousticness","Instrumentalness"];  
musicDataSet2 =  
    table(Dancibility,Energy,Valence,Acousticness,Speechiness,Instrumentalness,Genre)  
  
Warning: Column headers from the file were modified to make them valid  
MATLAB  
identifiers before creating variable names for the table. The original  
column  
headers are saved in the VariableDescriptions property.  
Set 'VariableNamingRule' to 'preserve' to use the original column  
headers as  
table variable names.
```

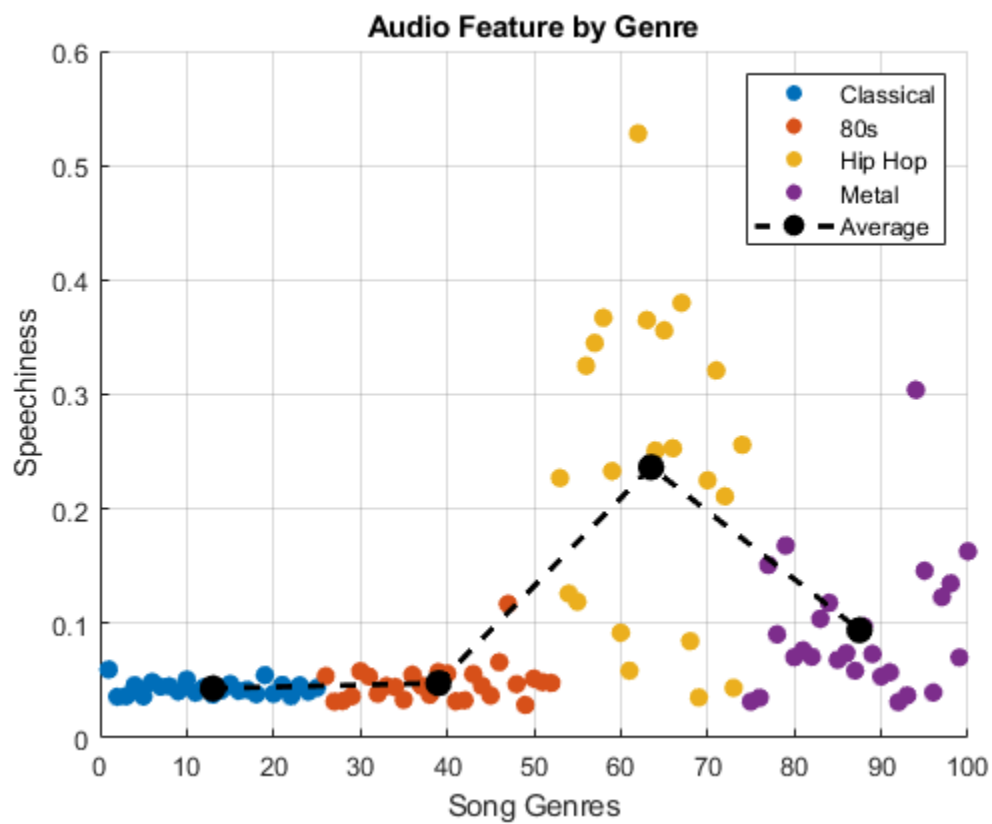
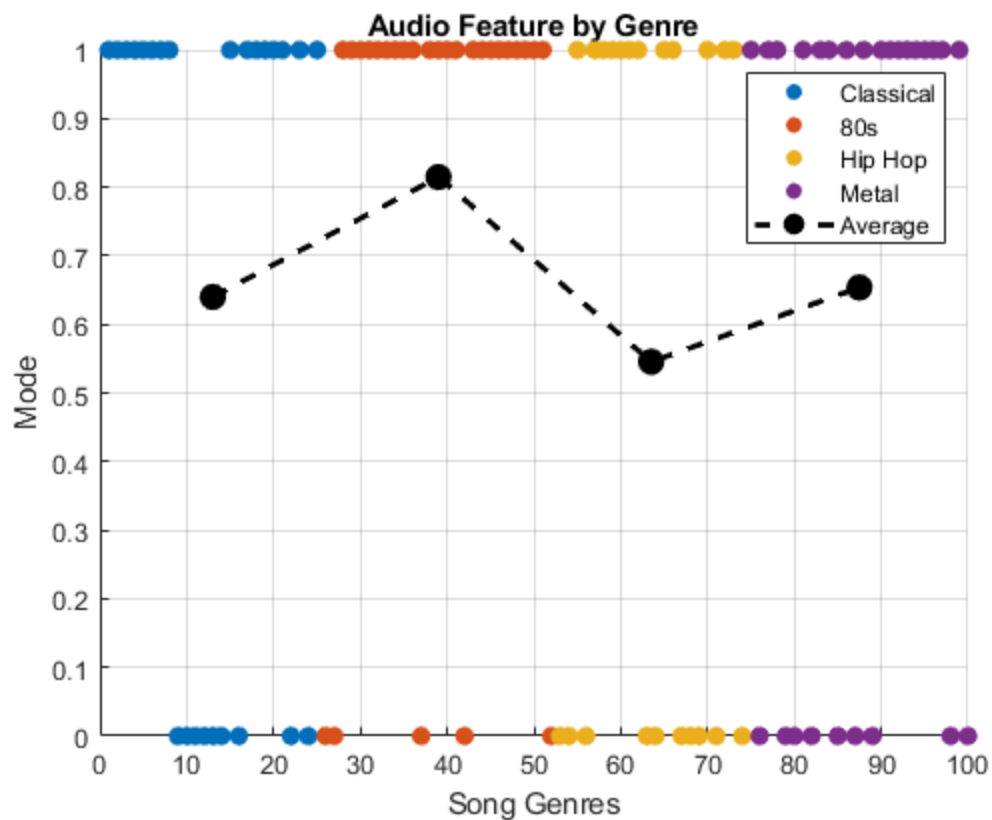
Plotting

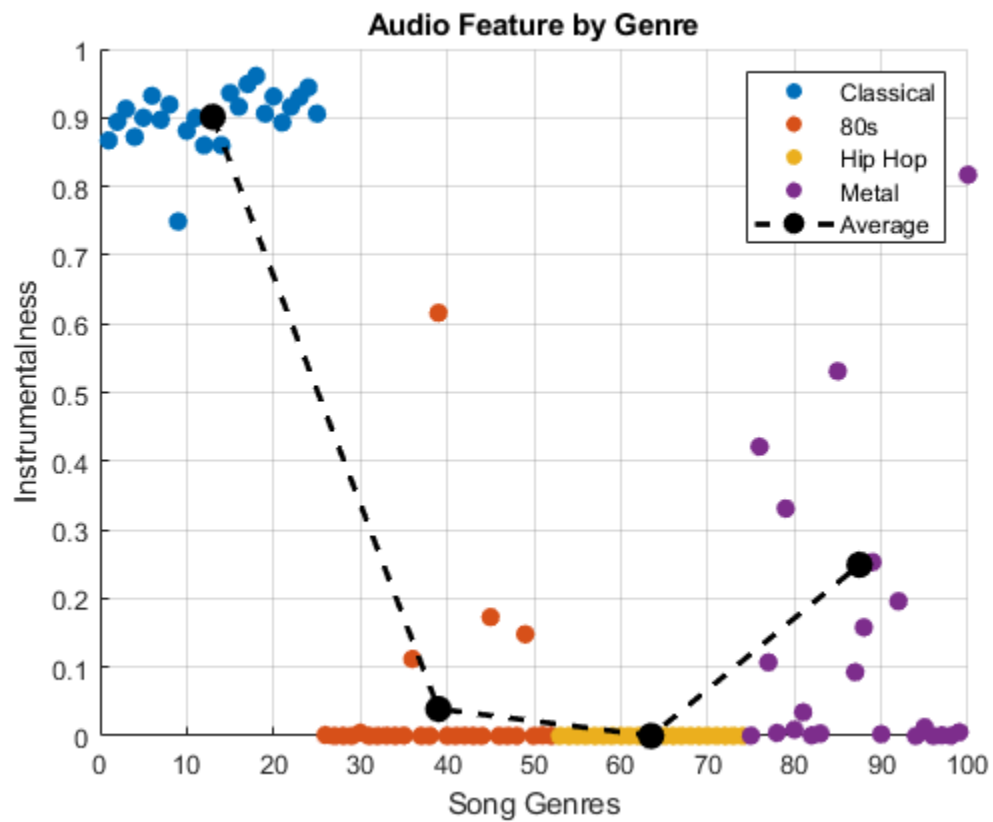
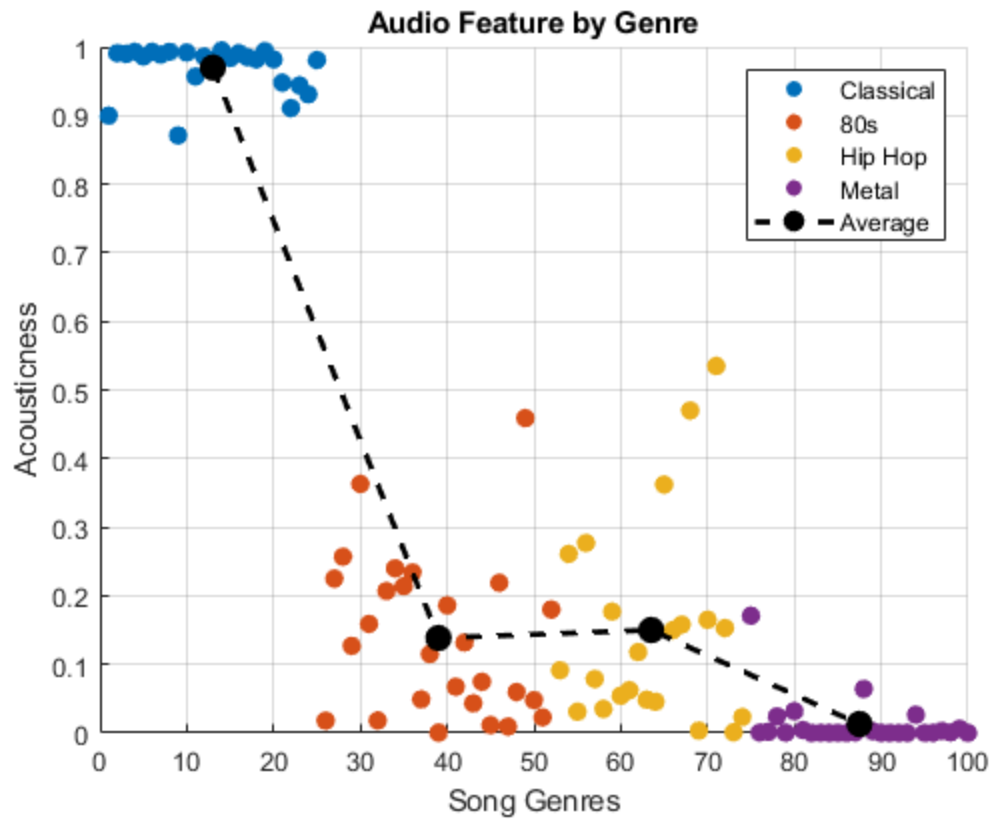
```
audioFeatBox = cell(1,width(audioFeaturesData));
for i = 1:width(audioFeaturesData)
    figure(i)
        for j = 1:numGenres
            genreSection = sum(songsPerGenre(1:j))+1 :
sum(songsPerGenre(1:j+1));
            songMean(j) = mean(genreSection);
            average(j) = mean(audioFeat4{j}(:,i));
            scatter(genreSection,audioFeat4{j}(:,i),50,'filled')
            hold on
        end
        plot(songMean,average,'--k.','Linewidth',2,'Markersize',34)
        hold on
        grid on
        xlabel('Song Genres')
        title('Audio Feature by Genre')
        ylabel(audioFeaturesLabels{i})
        if (i ==2 || i ==4) location = 'Southeast';
        else location = 'Northeast';end
        legend(['Classical' , "80s" , "Hip Hop" , "Metal"
, "Average"],'Location',location)
    end

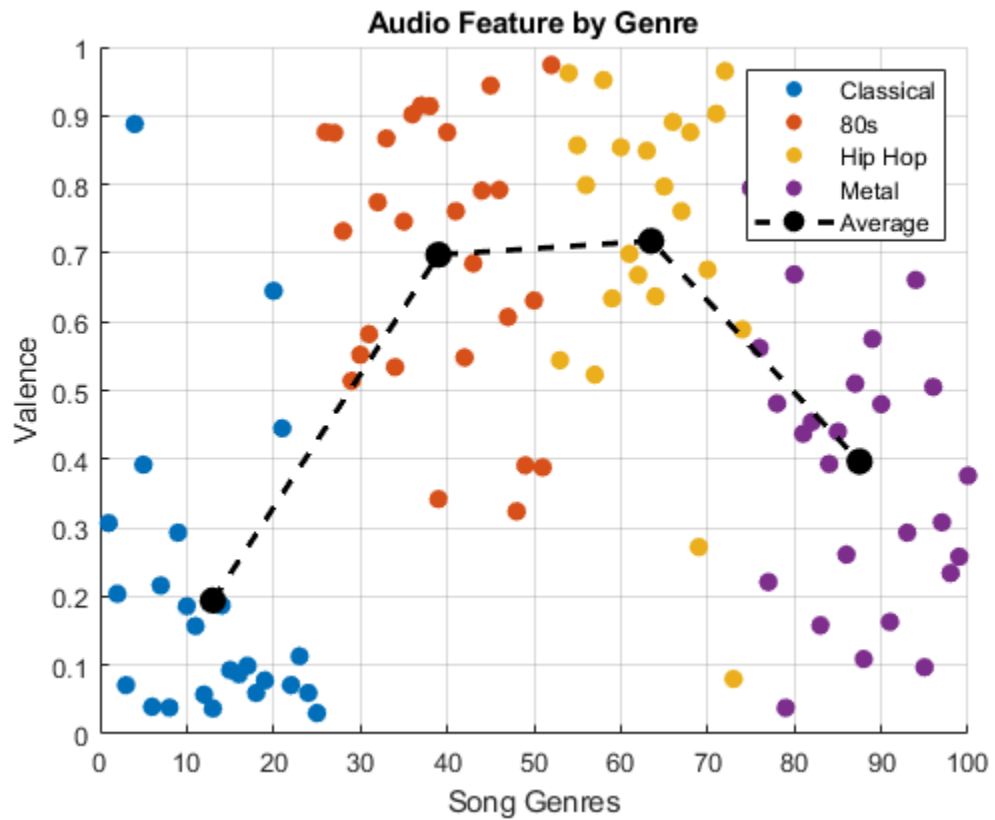
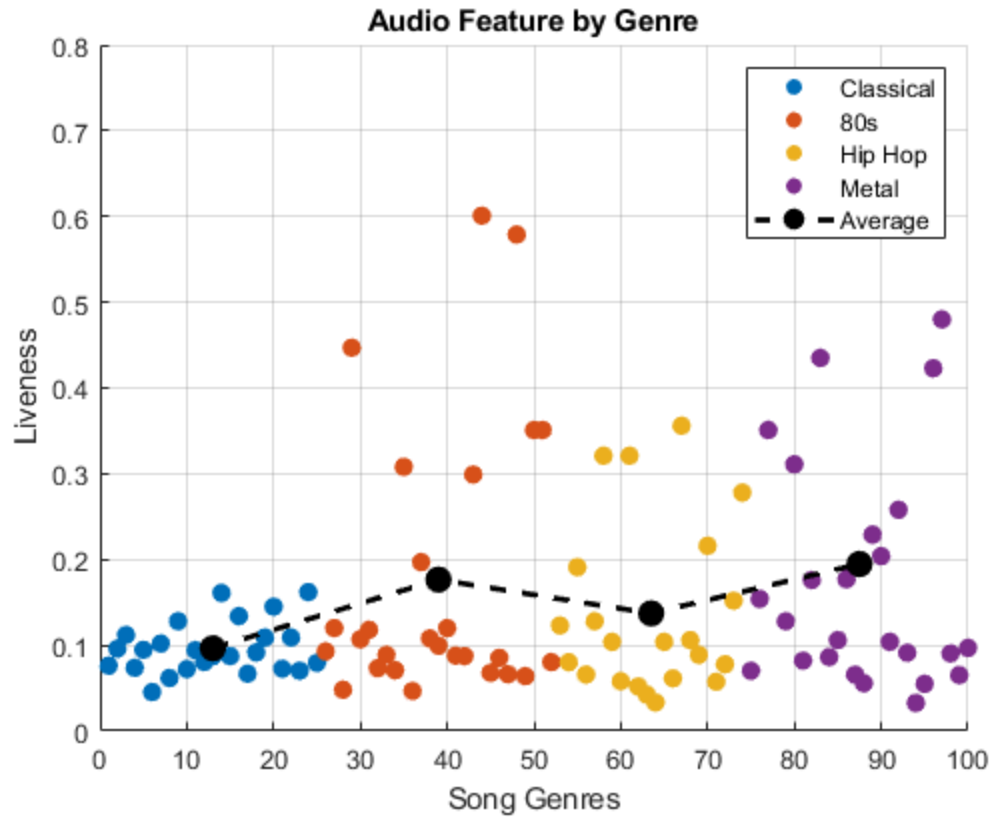
% for i = 1:width(audioFeaturesData)
%     figure(width(audioFeaturesData)+i)
%     boxplot(audioFeatBox{i},genreList)
%     title('Audio Feature by Genre')
%     hold on
%     grid on
%     xlabel('Song Genres')
%     ylabel(audioFeaturesLabels{i})
% end
```

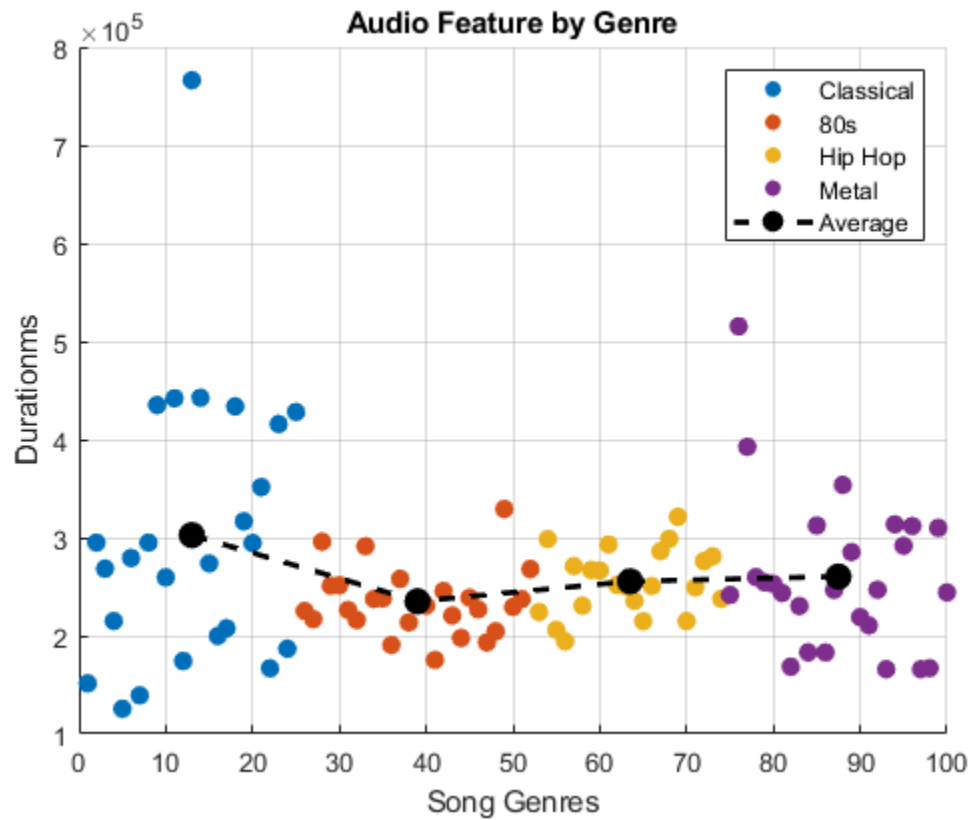
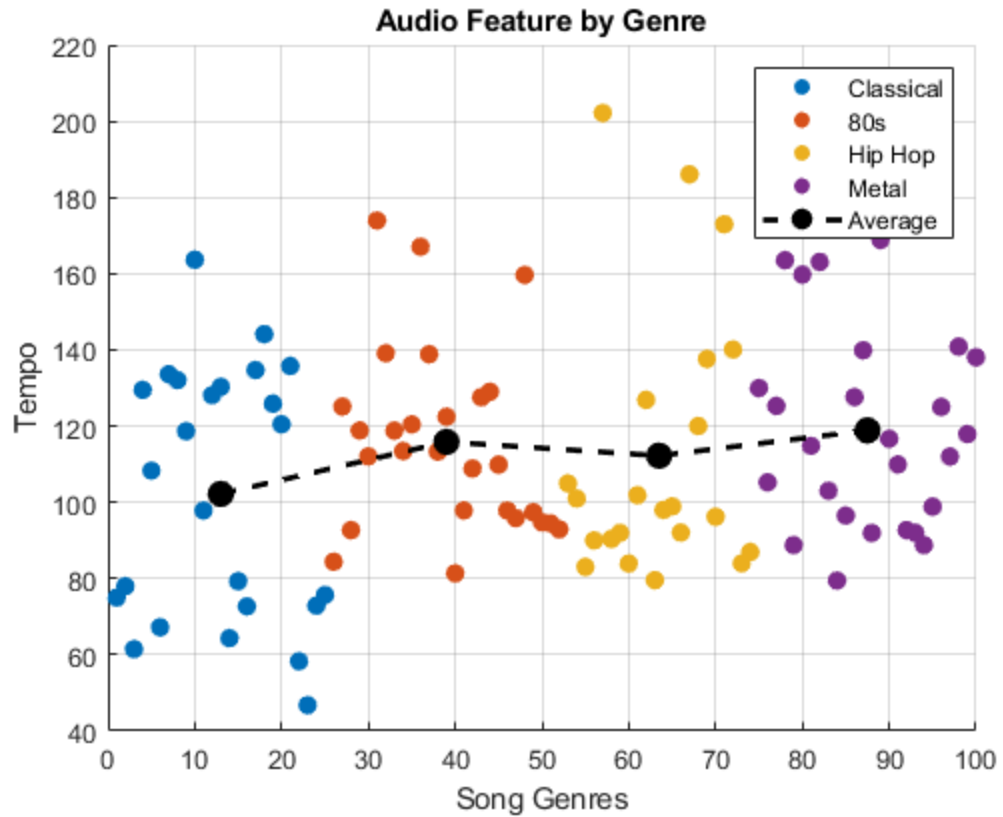












Importing TEST Data

```
audioFeaturesTEST = readtable('audioFeaturesNICK.csv');
trackInfoTEST = readtable('trackInfoNICK.csv');
trackInfoTEST(:,1) = [];

songsTEST = [1:height(audioFeaturesTEST)];
numSongsTEST = length(songsTEST);
songMeanTEST = zeros(numGenres,1);
averageTEST = zeros(numGenres,1);
genreTagsTEST = table2cell(trackInfoTEST(:,3));

GenreTEST = string(80*ones(numSongsTEST,1))+"s";

audioFeaturesDataTEST =
    audioFeaturesTEST(:,2:2:width(audioFeaturesTEST));
audioFeaturesDataTEST(:,end) = [];
audioFeaturesDataTEST(:,12:16) = [];

audioFeaturesDataTESTNew =
    zeros(height(audioFeaturesTEST),width(audioFeaturesDataTEST));
for i = 1:height(audioFeaturesTEST)
    for k=1:numGenres
        if (contains(genreTagsTEST{i},lower(genreList(k)))) == 1
            GenreTEST(i,1) = genreList(k); end
        end
        for j = 1 : width(audioFeaturesDataTEST)
            audioFeaturesDataTESTNew(i,j) =
                str2num(cell2mat(table2array(audioFeaturesDataTEST(i,j))));
        end
    end
end

songsPerGenreTEST = zeros(numGenres+1,1);
for i = 1:numGenres
    for j = 1:width(audioFeaturesDataTESTNew)
        audioFeat4TEST{i}(:,j) =
            audioFeaturesDataTESTNew(find(GenreTEST == genreList(i)),j);
    end
    songsPerGenreTEST(i+1) = height(audioFeat4TEST{i});
end

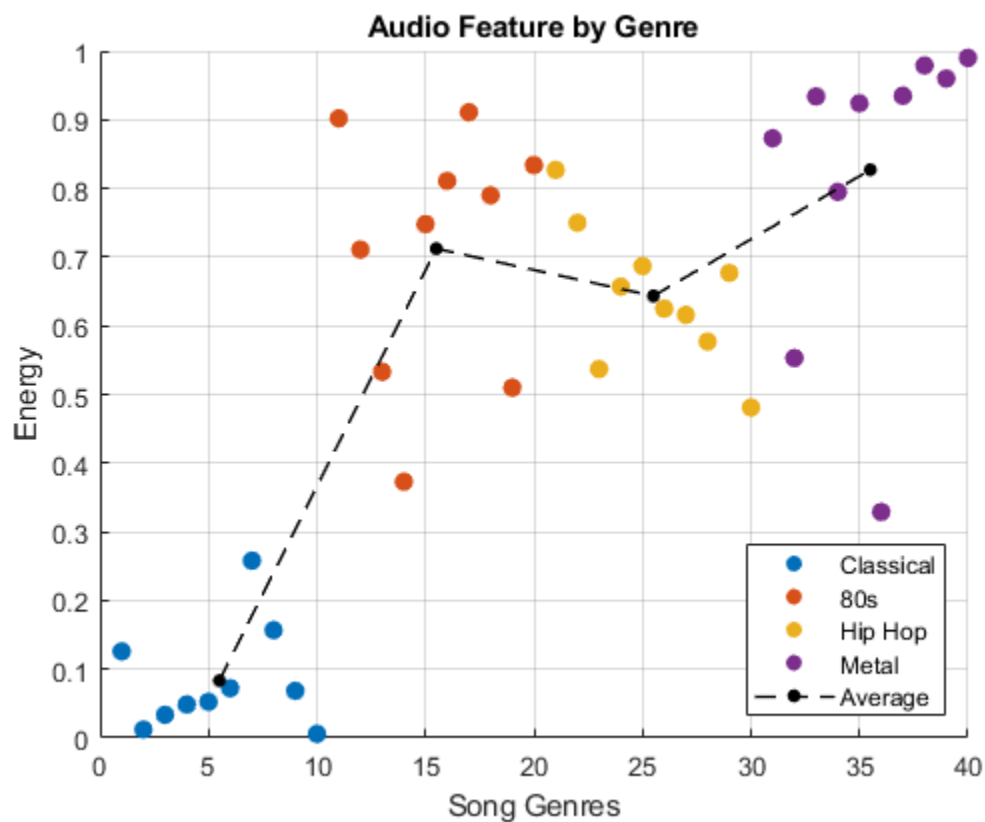
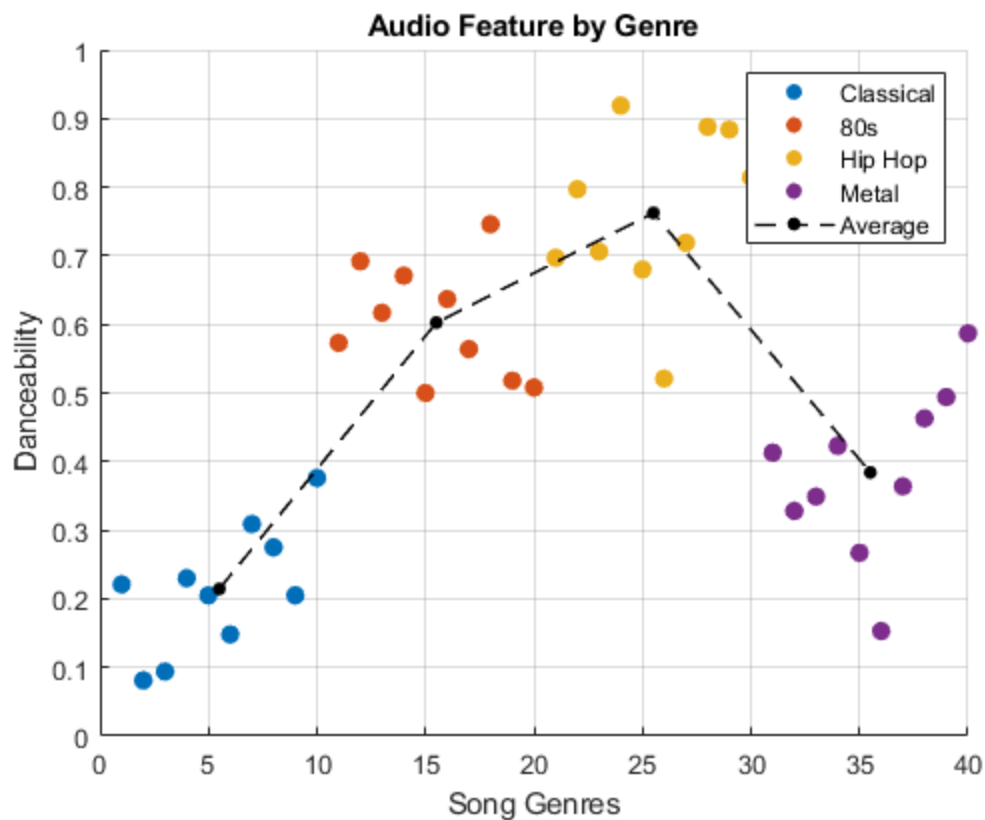
DancibilityTEST = audioFeaturesDataTESTNew(:,1);
EnergyTEST = audioFeaturesDataTESTNew(:,2);
KeyTEST = audioFeaturesDataTESTNew(:,3);
LoudnessTEST = audioFeaturesDataTESTNew(:,4);
modeTEST = audioFeaturesDataTESTNew(:,5);
SpeechinessTEST = audioFeaturesDataTESTNew(:,6);
AcousticnessTEST = audioFeaturesDataTESTNew(:,7);
InstrumentalnessTEST = audioFeaturesDataTESTNew(:,8);
LivenessTEST = audioFeaturesDataTESTNew(:,9);
ValenceTEST = audioFeaturesDataTESTNew(:,10);
TempoTEST = audioFeaturesDataTESTNew(:,11);
DurationTEST = audioFeaturesDataTESTNew(:,12);
```

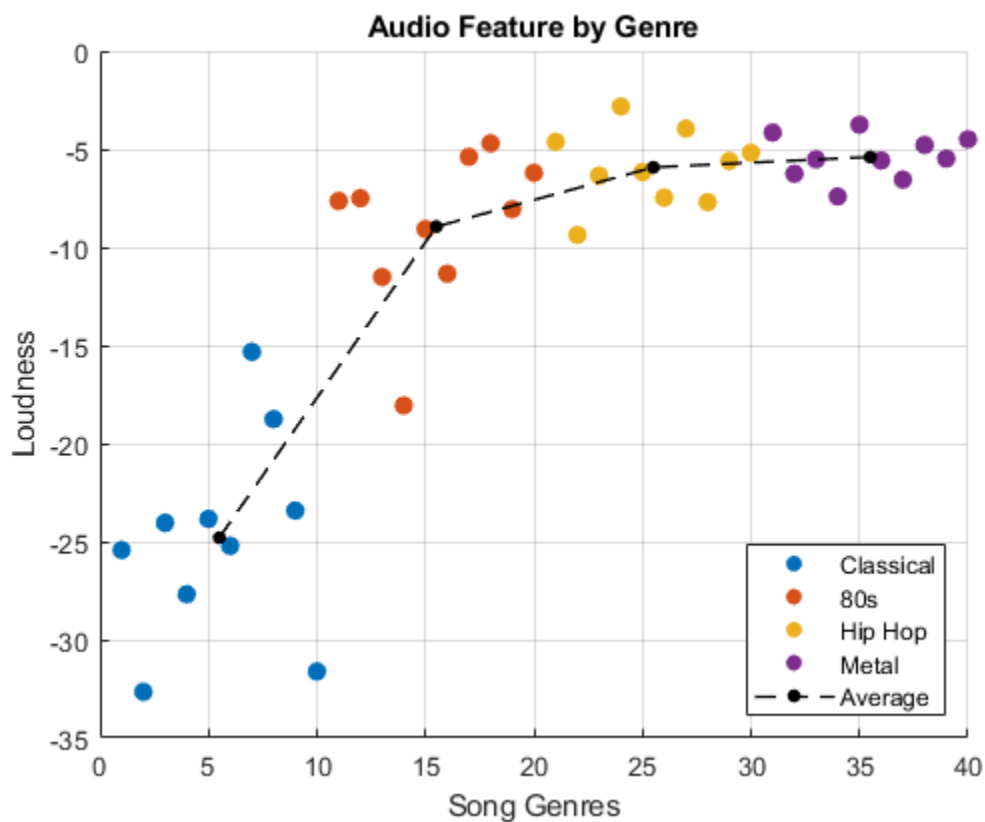
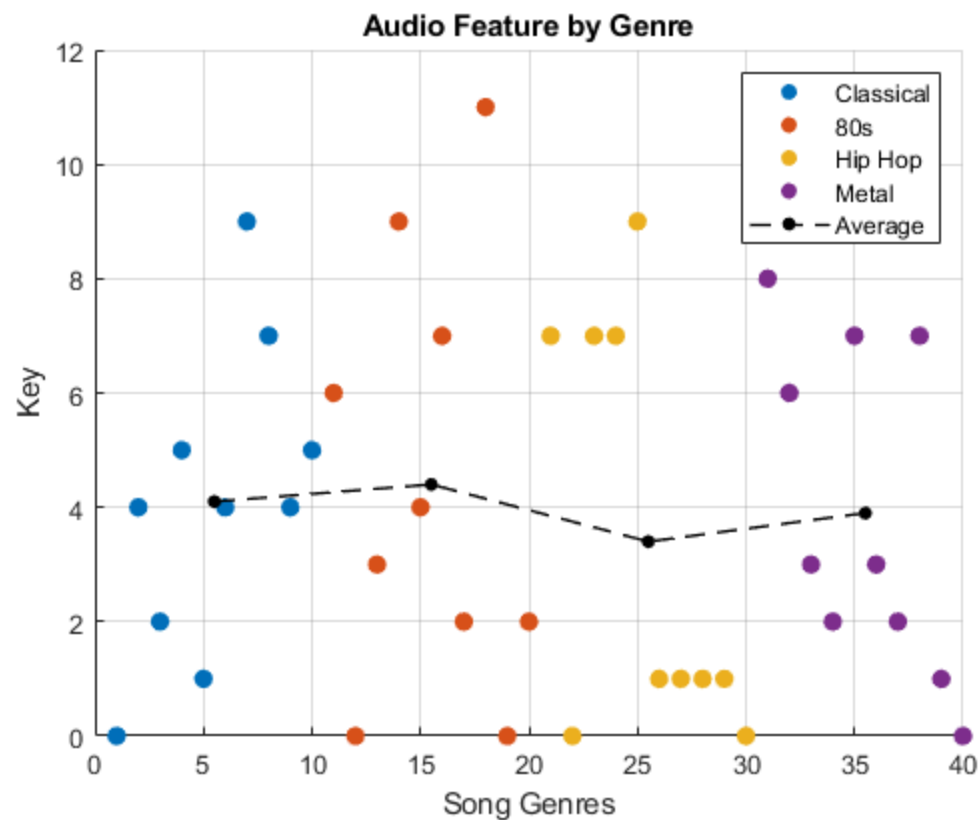
```
musicDataSetTEST = [DancibilityTEST EnergyTEST ValenceTEST  
    AcousticnessTEST InstrumentalnessTEST];  
musicDataSetTEST2 = [DancibilityTEST EnergyTEST ValenceTEST  
    AcousticnessTEST SpeechinessTEST InstrumentalnessTEST];
```

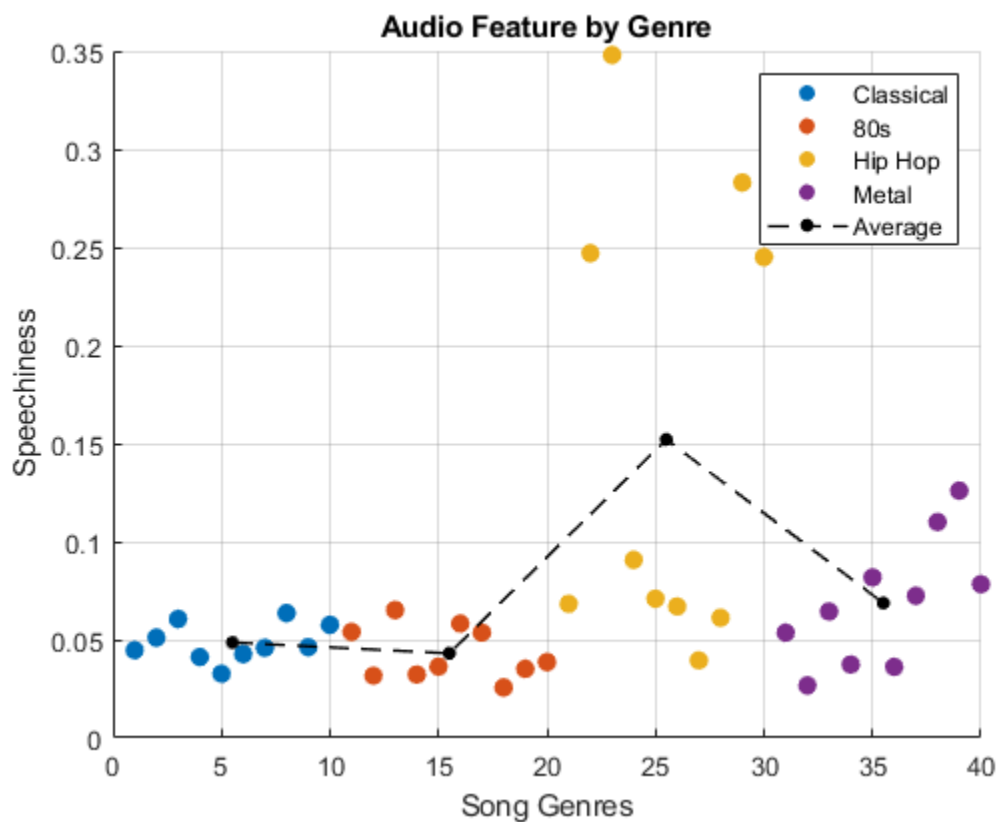
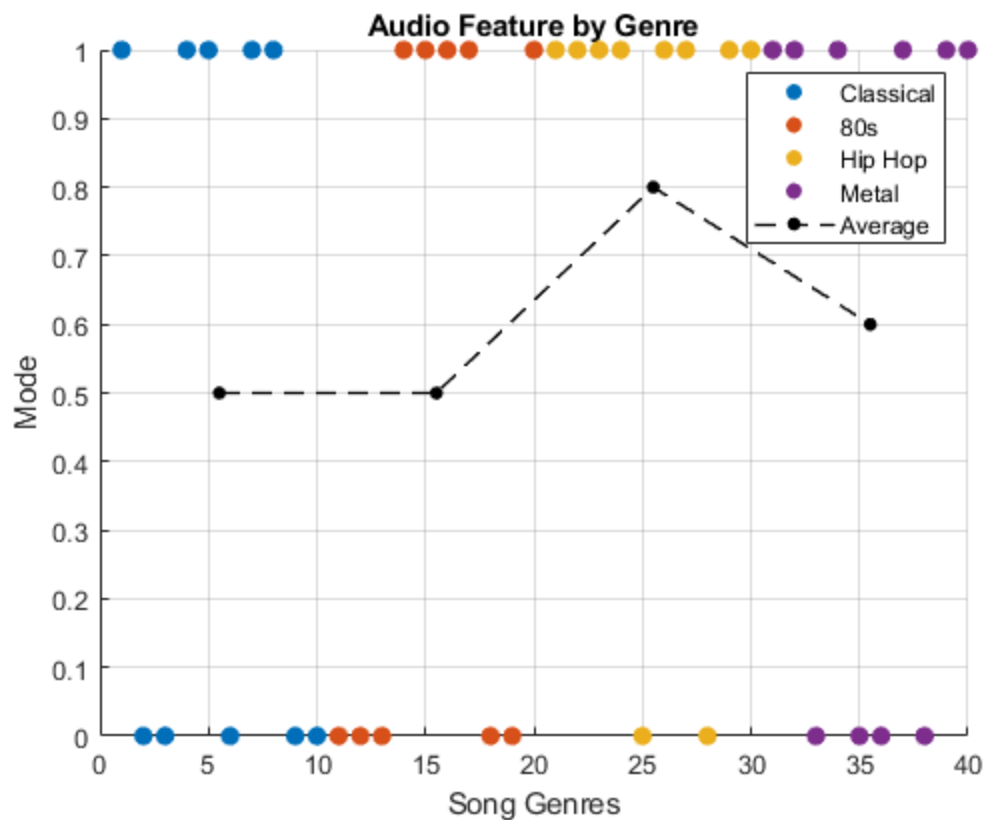
*Warning: Column headers from the file were modified to make them valid
MATLAB
identifiers before creating variable names for the table. The original
column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
headers as
table variable names.*

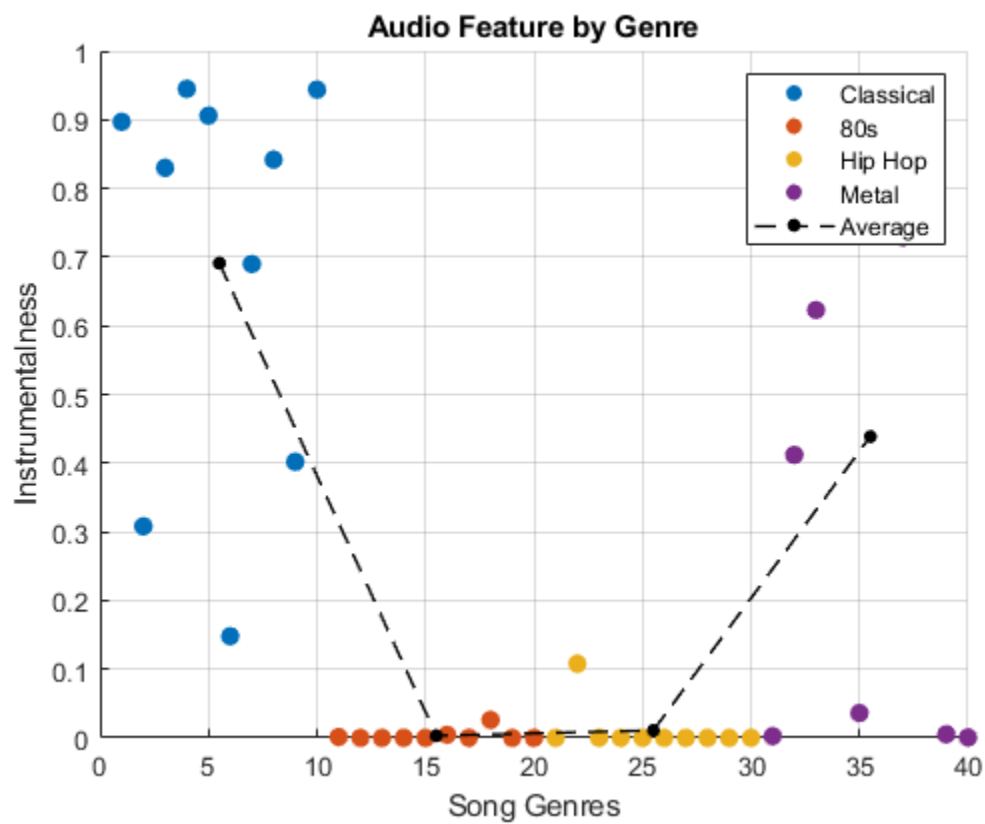
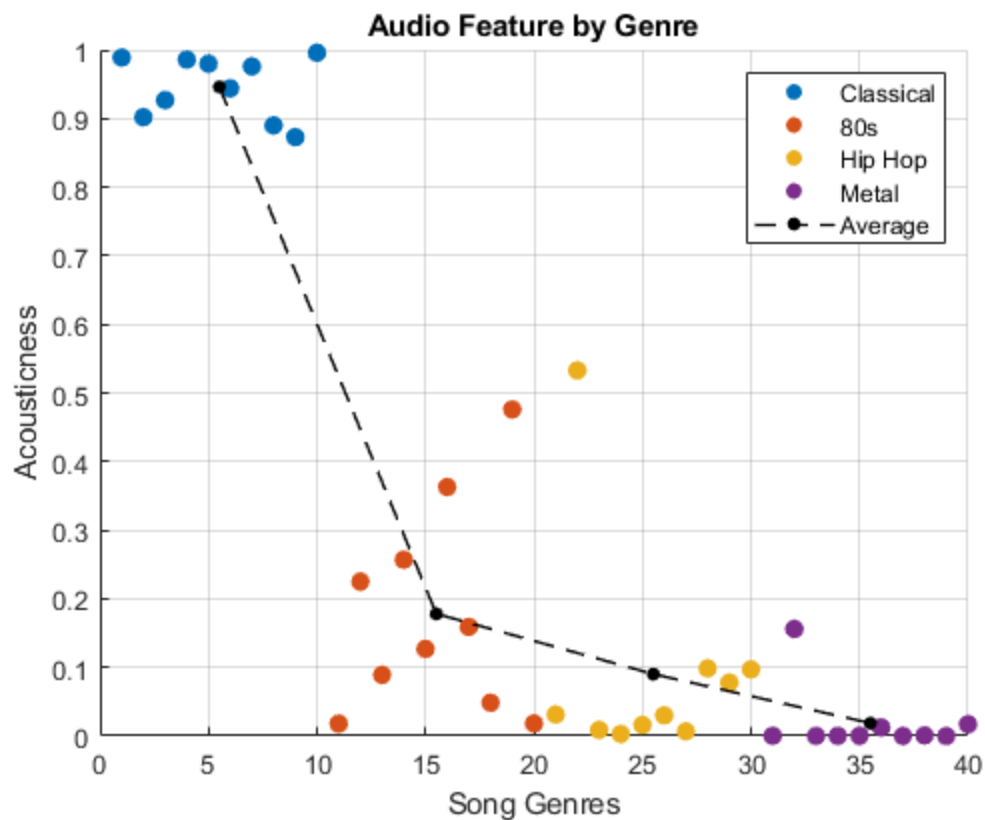
Plotting

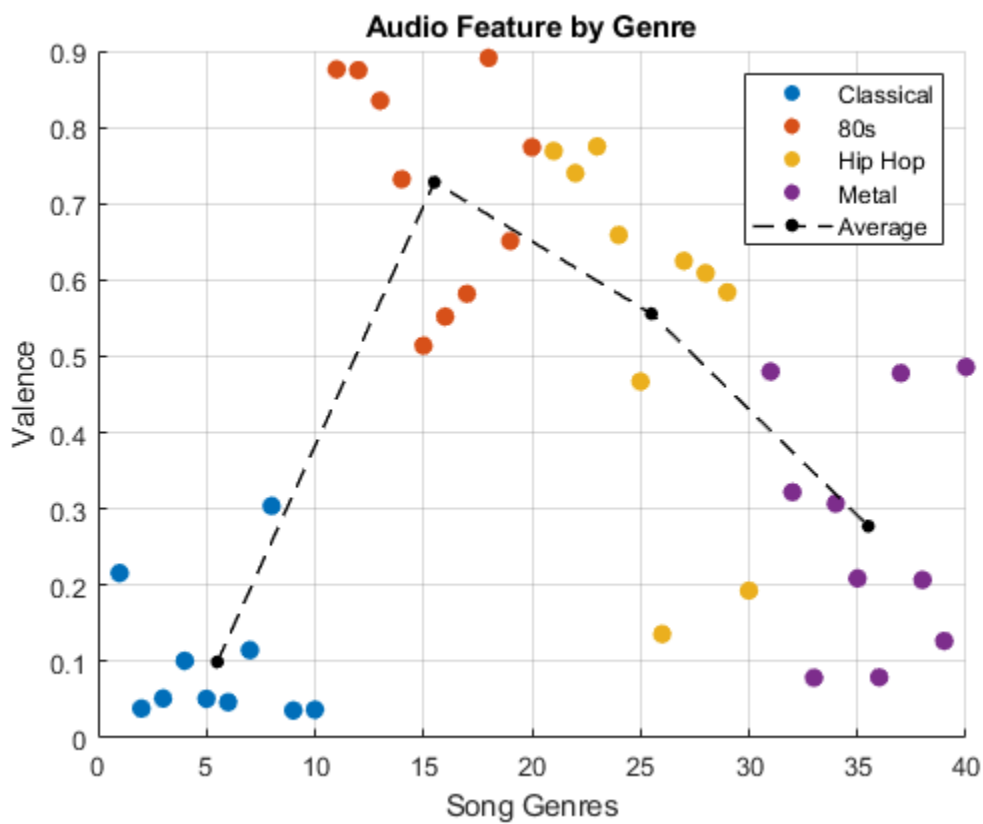
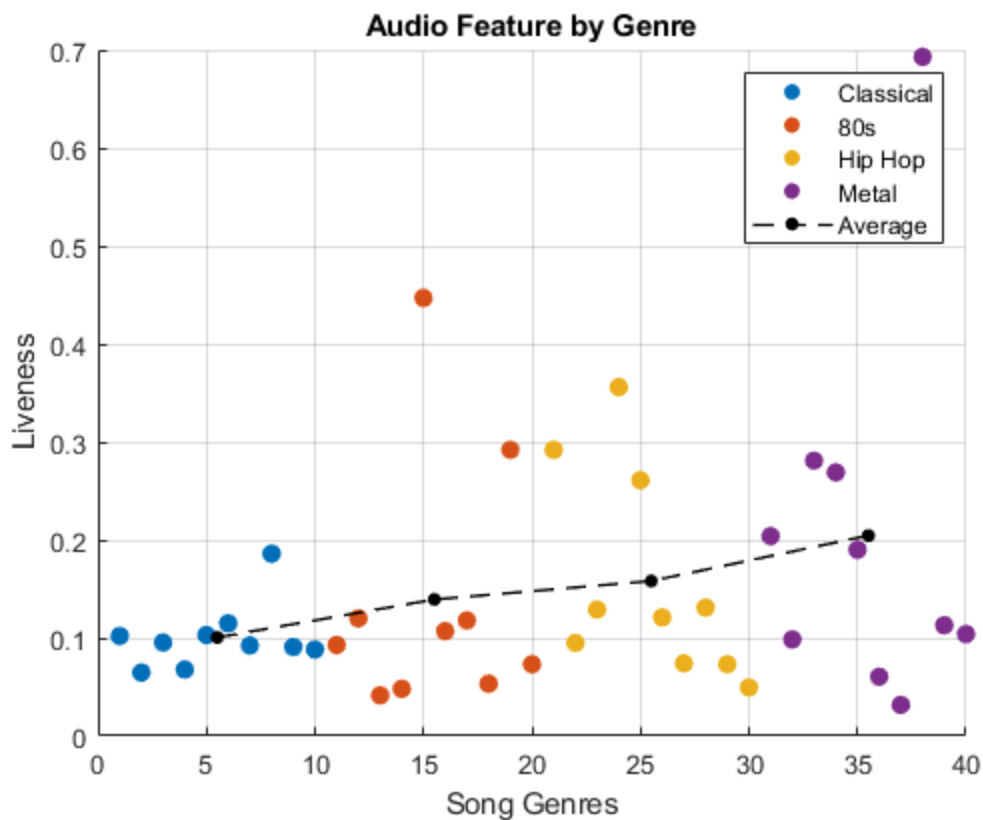
```
for i = 1:width(audioFeaturesDataTEST)  
    figure(i+12)  
        for j = 1:numGenres  
            genreSection = sum(songsPerGenreTEST(1:j))+1 :  
sum(songsPerGenreTEST(1:j+1));  
            songMeanTEST(j) = mean(genreSection);  
            averageTEST(j) = mean(audioFeat4TEST{j}(:,i));  
            scatter(genreSection, audioFeat4TEST{j}(:,i), 50, 'filled')  
            hold on  
        end  
        plot(songMeanTEST, averageTEST, '--  
k.', 'Linewidth', 1, 'Markersize', 17)  
        hold on  
        grid on  
        xlabel('Song Genres')  
        title('Audio Feature by Genre')  
        ylabel(audioFeaturesLabels{i})  
        if (i == 2 || i == 4) location = 'Southeast';  
        else location = 'Northeast'; end  
        legend(["Classical" , "80s" , "Hip Hop" , "Metal"  
    , "Average"], 'Location', location)  
    end
```

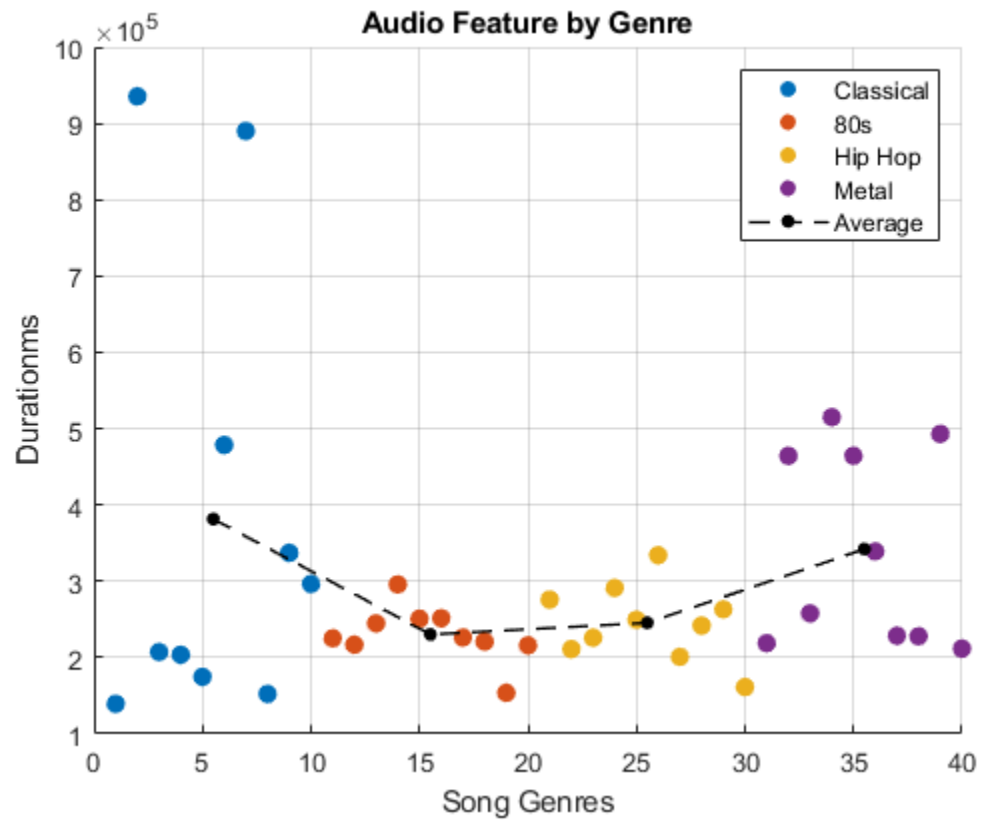
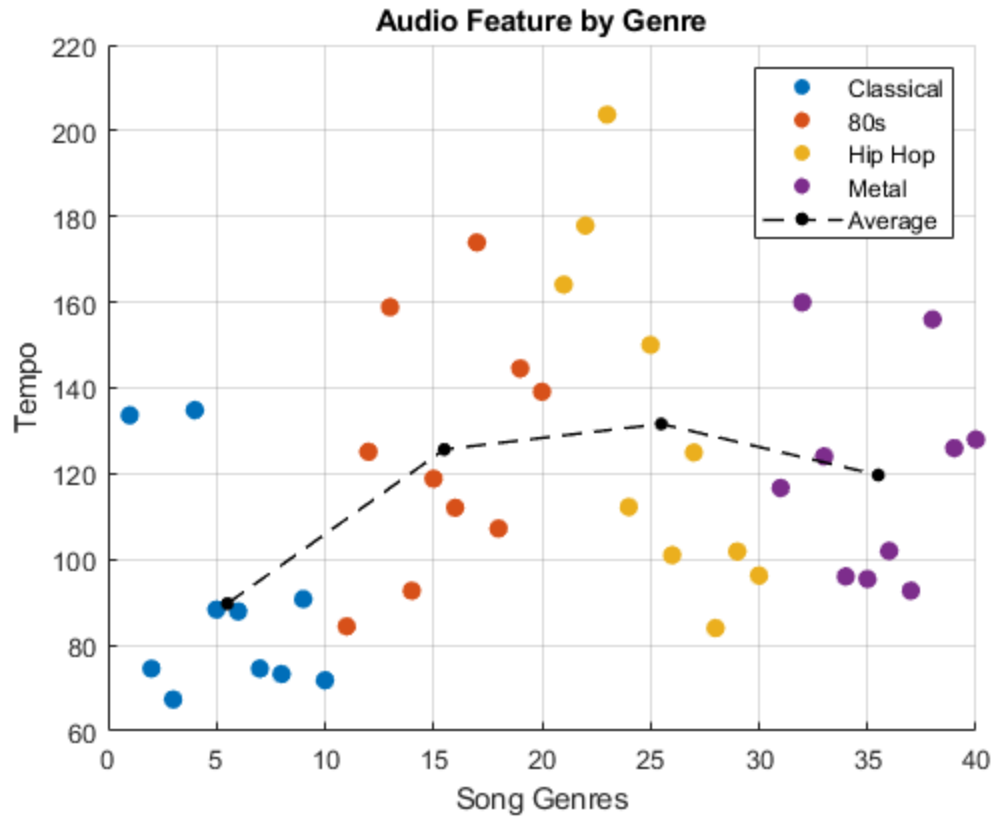












KNN Model

```
maxNeighbors = 40;
distIter = ['minkowski', 'chebychev'];
PredictedTrain = cell(maxNeighbors,1);
PredictedTest = cell(maxNeighbors,1);
AccuracyTrain = zeros(maxNeighbors,1);
AccuracyTest = zeros(maxNeighbors,1);
for k = 1:maxNeighbors
    model = fitcknn(musicDataSet(:,1:end-1),musicDataSet(:,end));
    model.NumNeighbors = k;
    model.Distance = 'chebychev';
    PredictedTrain{k} = string(predict(model,musicDataSet));
    PredictedTest{k} = string(predict(model,musicDataSetTEST));

    Error(k) = resubLoss(model);
    AccuracyTrain(k) = 100*(sum(PredictedTrain{k} == Genre)/
length(PredictedTrain{k}));
    AccuracyTest(k) = 100*(sum(PredictedTest{k} == GenreTEST)/
length(PredictedTest{k}));
end

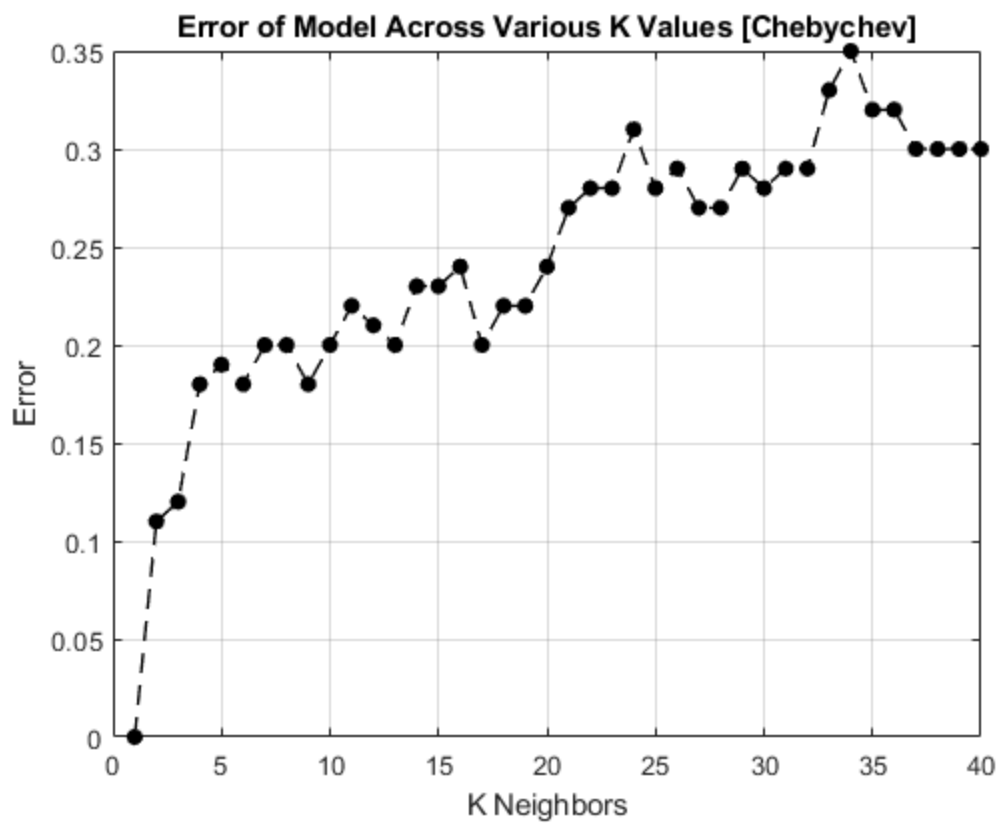
figure(25)
plot(1:maxNeighbors,Error,'--k.','Linewidth',1,'Markersize',21)
hold on
grid on
xlabel('K Neighbors')
ylabel('Error')
title('Error of Model Across Various K Values [Chebychev]')

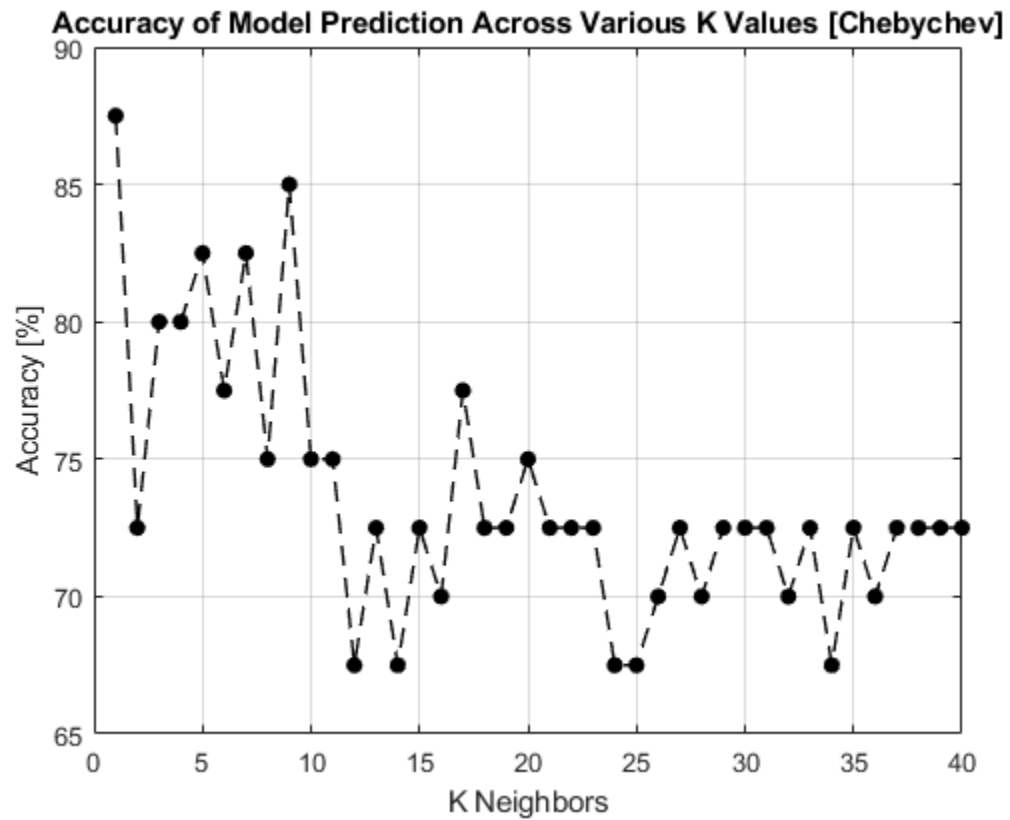
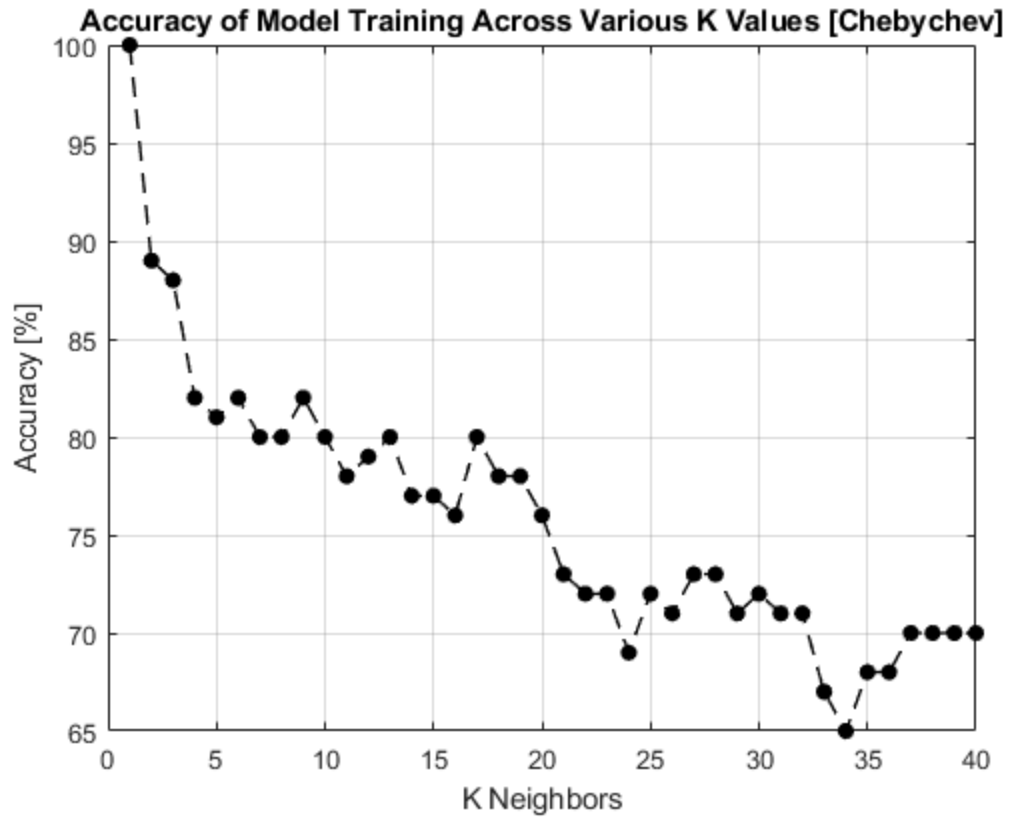
figure(26)
plot(1:maxNeighbors,AccuracyTrain,'--
k.','Linewidth',1,'Markersize',21)
hold on
grid on
xlabel('K Neighbors')
ylabel('Accuracy [%]')
title('Accuracy of Model Training Across Various K Values
[Chebychev]')

figure(27)
plot(1:maxNeighbors,AccuracyTest,'--
k.','Linewidth',1,'Markersize',21)
hold on
grid on
xlabel('K Neighbors')
ylabel('Accuracy [%]')
title('Accuracy of Model Prediction Across Various K Values
[Chebychev]')

disp("Best Model Accuracy (" + string(3) + " neighbors) = " +
string(max(AccuracyTest)) + "%")
```

Best Model Accuracy (3 neighbors) = 87.5%





Published with MATLAB® R2020b