Changes You Can Make


AI Model Music Example: Legends vs. Newcomers

Table of Contents

Step 1: Define Success
Step 2: Import Your Music Data
Step 3: Identifying Your Variables
Step 4: Split Data into Training & Testing Sets
Step 5: Finding Your Model
Step 6: Fitting a Model to Your Data
Step 7: Run Model Using Your Test Dataset
Step 8: Evaluate the Model
Step 9: Using the Model

Step 1: Define Success

In this example, we'll be using the albums of three successful rappers (Eminem, Jay-Z, and Kendrick Lamar) and Veeze, a rapper who released his debut studio album in 2023. Our goal is to compare the albums of the three successful rappers to Veeze's debut album to determine if he will

Introduction

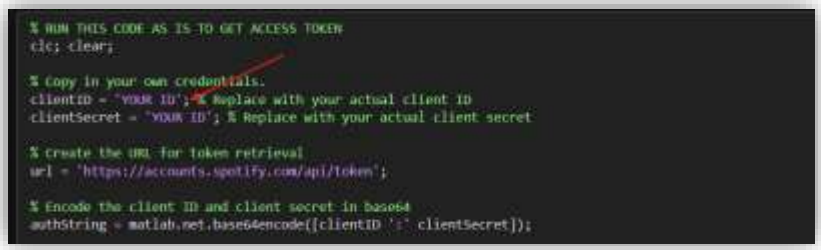
Today we will review how you can make the KNN model music example in the live script **'RapMusic_ProjectExample.mlx'** into your own unique AI model. To achieve this, you only need to make 3 major changes to the code:

1. Find new artists
2. Select new song features other than duration and danceability
3. Train your model and adjust its accuracy

Additional changes might be necessary, but you mainly focus on the changes listed above.

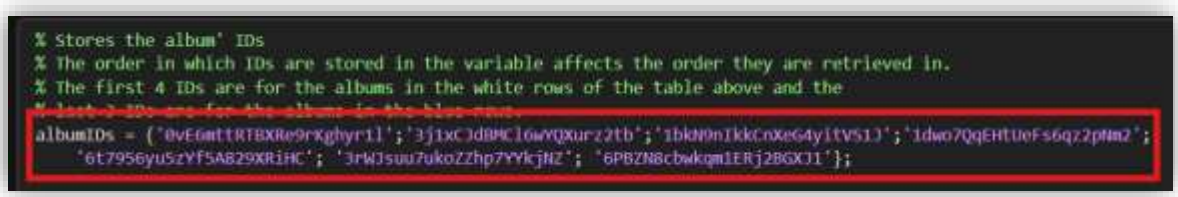
Access Token

Before discussing the first change it is important to note that no change is necessary to code use to retrieve the **Access Token**. The only modification in this section will paste in your **Client ID** and **Client Secret** from Spotify.

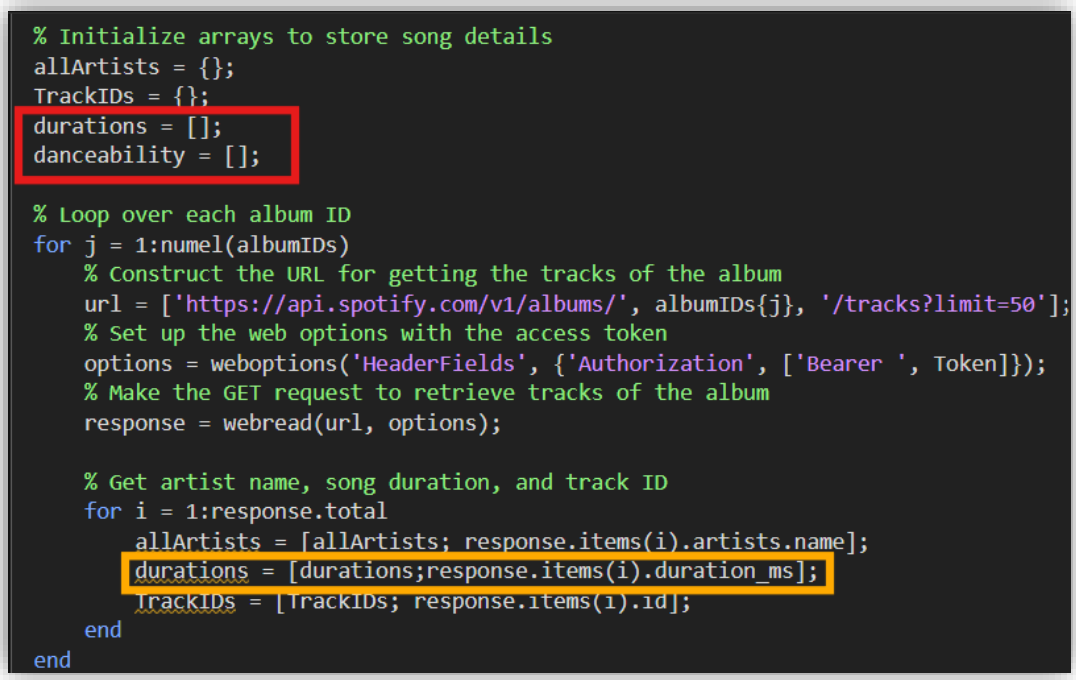

```
% RUN THIS CODE AS IS TO GET ACCESS TOKEN
clc; clear;

% Copy in your own credentials:
clientID = 'YOUR ID'; % Replace with your actual client ID
clientSecret = 'YOUR ID'; % Replace with your actual client secret

% Create the URL for token retrieval
url = 'https://accounts.spotify.com/api/token';

% Encode the client ID and client secret in base64
authString = matlab.net.base64encode([clientID ':' clientSecret]);
```

Find New Artists

The first major change is to choose new artists to compare. Select new artists and find at least two albums for each. Once you have decided which albums you are going to use, look up the **Album ID** for each album and paste them into the variable `albumIDs`. Use the same process you used in Section 03 to find **Artists IDs** to find the **Album IDs.**



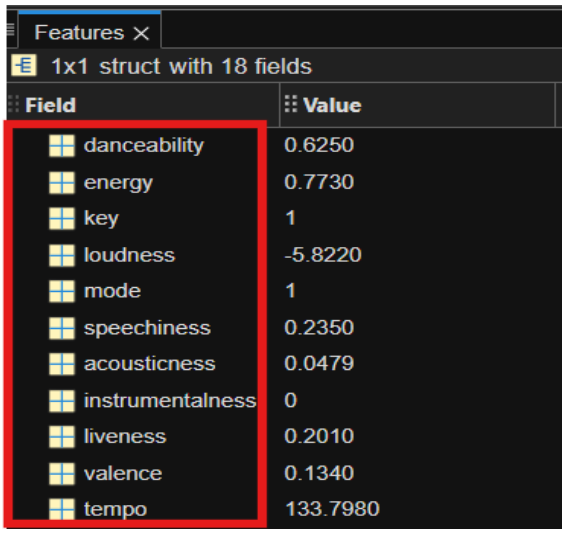
No changes are required to the first **for loop**, but you will need to rename the variables in the red rectangle in the image below. You can rename them once you have decided what song features you are going to use to train the KNN model in the next step. Also, delete the line of code in the yellow rectangle.



Select New Parameters

Now you need to select new song features for your model. The example used **song duration** and **danceability**, but there are many song audio features you can choose from. The image on the right shows the parameters you can choose from. All of these values are stored within the variable `Features`. After selecting the new parameters, you will need to update the code below with the correct dot notation. For example, if you chose **tempo** and **energy** the new code would be:

```
tempo =  [tempo; Features.tempo];

Energy = [energy; Features.energy];.
```



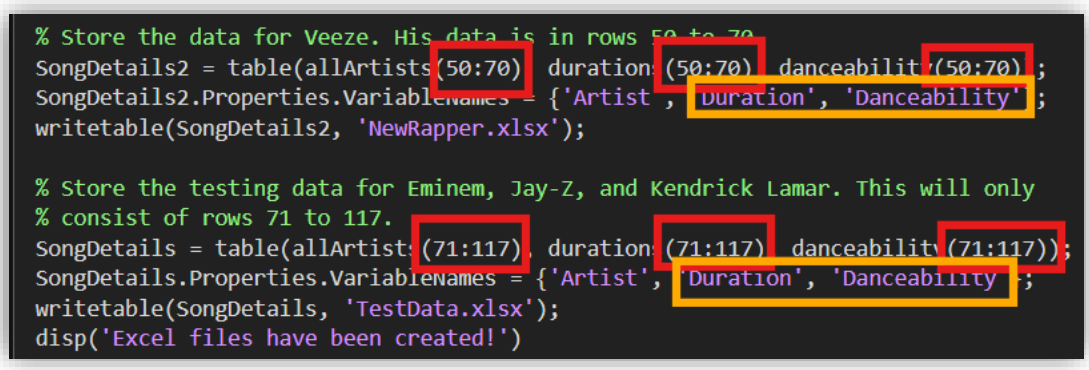

```
% Loop over each track to get its audio features
for k = 1:numel(TrackIDs)
    url2 = ['https://api.spotify.com/v1/audio-features/', TrackIDs{k}];
    options2 = weboptions('HeaderFields', {'Authorization', ['Bearer ', Token]});
    Features = webread(url2, options2);
    danceability = [danceability;Features.danceability];
end
```

Minor Modifications

Some minor modifications will be necessary before continuing. First, examine the variable `allArtists` to find which rows contain more than one artist name. Fix each of those rows, so there is only one artist. Make sure you change the names to the new artists you selected.

```
% Some songs were collaborations so they include more names than one.
% We only want the variable allArtists to contain the names of the main
% artists: 'Eminem','Jay-Z', 'Kendrick Lamar, & 'Veeze'.
allArtists(1:20) = {'Eminem'};
allArtists(71:88) = {'Eminem'};
allArtists(21:34) = {'Jay-Z'};
allArtists(89:104) = {'Jay-Z'};
allArtists(35:49) = {'Kendrick Lamar'};
allArtists(105:117) = {'Kendrick Lamar'};
allArtists(50:70) = {'Veeze'};
```

Additionally, make sure that you are splitting and storing the data correctly. You will need to adjust the row numbers (red rectangles) in the code below to ensure that the data is separated correctly. Also, make sure to rename the columns (yellow rectangles) of the new table you are creating to match the new parameters you selected.



Train Your Model

Lastly, train your model using `fitcknn()`. You can also delete the line of code in the red rectangle in the image below since it is not needed. Now you should be able to run the remaining steps without making any further changes. If you need to improve the accuracy of your model, try adjusting the **NumNeighbors** option in the function `fitcknn()` or try to run the code without using the NumNeighbors option. For example, you can try using `fitcknn (Rappers3, "Artist")`.