recommendation

December 11, 2022

[ ]:
```
In the code below, I used the token that I gained from the authorization Python␣
↪script as a means to access data from Spotify. The result of this code was␣
↪the table that you see in music.csv which consists of Spotify songs from all␣
↪of the genres and their corresponding valence and energy measurements in the␣
↪Spotify database.
```

[1]:
```
import pandas as pd
import sys
import authorization
import pandas as pd
from tqdm import tqdm
import time
import numpy as np
from numpy.linalg import norm
```

[ ]:
```
The variable "sp" is the token that my authorization python script returns␣
↪which allows me to access data on Spotify.
```

[2]:
```
sp = authorization.authorize()
```

[ ]:
```
In the code below, I returned the number of listed genres on Spotify and a list␣
↪of these genres.
```

[3]:
```
genres = sp.recommendation_genre_seeds()
```

[6]:
```
print(len(genres))
```

```
126
```

[7]:
```
print(genres)
```

```
['acoustic', 'afrobeat', 'alt-rock', 'alternative', 'ambient', 'anime', 'black-
metal', 'bluegrass', 'blues', 'bossanova', 'brazil', 'breakbeat', 'british',
'cantopop', 'chicago-house', 'children', 'chill', 'classical', 'club', 'comedy',
'country', 'dance', 'dancehall', 'death-metal', 'deep-house', 'detroit-techno',
'disco', 'disney', 'drum-and-bass', 'dub', 'dubstep', 'edm', 'electro',
'electronic', 'emo', 'folk', 'forro', 'french', 'funk', 'garage', 'german',
'gospel', 'goth', 'grindcore', 'groove', 'grunge', 'guitar', 'happy', 'hard-
rock', 'hardcore', 'hardstyle', 'heavy-metal', 'hip-hop', 'holidays', 'honky-
tonk', 'house', 'idm', 'indian', 'indie', 'indie-pop', 'industrial', 'iranian',
'j-dance', 'j-idol', 'j-pop', 'j-rock', 'jazz', 'k-pop', 'kids', 'latin',
'latino', 'malay', 'mandopop', 'metal', 'metal-misc', 'metalcore', 'minimal-
techno', 'movies', 'mpb', 'new-age', 'new-release', 'opera', 'pagode', 'party',
'philippines-opm', 'piano', 'pop', 'pop-film', 'post-dubstep', 'power-pop',
'progressive-house', 'psych-rock', 'punk', 'punk-rock', 'r-n-b', 'rainy-day',
'reggae', 'reggaeton', 'road-trip', 'rock', 'rock-n-roll', 'rockabilly',
'romance', 'sad', 'salsa', 'samba', 'sertanejo', 'show-tunes', 'singer-
songwriter', 'ska', 'sleep', 'songwriter', 'soul', 'soundtracks', 'spanish',
'study', 'summer', 'swedish', 'synth-pop', 'tango', 'techno', 'trance', 'trip-
hop', 'turkish', 'work-out', 'world-music']
```

[ ]:
```
In the code below, the data_dict shows the data that I collected from each song␣
↪in Spotify. For all the 126 genres listed above, I got 50 songs from genre␣
↪and put the data from each of these songs into the categories in data_dict.
```

[12]:
```
data_dict = {"id":[], "genre":[], "track_name":[], "artist_name":[],
             "valence":[], "energy":[]}
```

[13]:
```
for g in tqdm(genres):

    recs = sp.recommendations(genres = [g], limit = 50)
    recs = eval(recs.json().replace("null", "-999").replace("false", "False").
 ↪replace("true", "True"))["tracks"]

    for track in recs:
        data_dict["id"].append(track["id"])
        data_dict["genre"].append(g)
        track_meta = sp.track(track["id"])
        data_dict["track_name"].append(track_meta.name)
        data_dict["artist_name"].append(track_meta.album.artists[0].name)
        track_features = sp.track_audio_features(track["id"])
        data_dict["valence"].append(track_features.valence)
        data_dict["energy"].append(track_features.energy)

        time.sleep(0.2)
```

```
100%|                      | 126/126 [45:27<00:00, 21.64s/it]
```

[ ]:
```
Using data_dict, I created a pandas dataframe which I used to create the music␣
↪csv file that shows all of the song data that I gathered from Spotify. This␣
↪table of song data represented the set of songs that would be potentially␣
↪recommended based on the Spotify song that I input. This code was done in␣
↪the Application Python journal.
```

[29]:
```
df = pd.DataFrame(data_dict)
df.to_csv("music.csv")
```

[ ]:
```

```