Anexo V. Clasificación de las canciones en clásicas o no clásicas

A continuación se presenta el código empleado para dividir nuestro conjunto de canciones en dos: si se consideran clásicas o no.

Para ello hemos usado el siguiente criterio:

- Si el principal artista de la canción tiene asignado el género “classical” o “classical era”, dicho artista se considerará como artista clásico.
- En caso de que aparezca cualquier otro género, dicho artista será considerado no clásico.

Atención: este documento contiene información personal para la autentificación del programa en Spotify. Bajo la asunción de que este programa se presenta con fines académicos se restringe su publicación o reparto. Los únicos usuarios autorizados para compartir esta información son los integrantes del grupo y las docentes de la asignatura Proyecto II, Integración de datos 2020 - 2021.

```
import re
import sys
import spotipy
import pandas as pd
from time import process_time
from spotipy.oauth2 import SpotifyClientCredentials

########################################################################
#                   ORDEN A EJECUTAR EN TERMINAL                       #
########################################################################

"""cd /root/work/Adquisición
   python3 artist_ids.py"""



#CARGAMOS EL CONJUNTO DE DATOS EN UN DATAFRAME Y PREPARAMOS EL CLIENTE PARA LA API

output = pd.read_csv("/root/work/Tratado/outputpopart.csv")
clientID, clientSec = '0d2beefe0f0846eea477a529f194cad5',
'5fdcae7eaa6e420d932ecb01a9506bd8'
ccm = SpotifyClientCredentials(client_id=clientID, client_secret=clientSec)
sp = spotipy.Spotify(client_credentials_manager=ccm)

#VAMOS A GUARDARNOS EN UN DICCIONARIO SI LOS ARTISTAS SON CLÁSICOS O NO Y LA FILA
POR LA CUAL EMPEZAR
```

```
classical = {}

try:
    with open("/root/work/Outputs/traceback.txt","r") as f:
        start = int(f.readline())
except FileNotFoundError:
    start = 0


#PROGRAMA PRINCIPAL

print(f"Empezando desde la fila {start}")

t1 = process_time()
for i in range(start, len(output)+1):
    try:
        to_re_id = output["artist_ids"][i]
        if not pd.isna(to_re_id):
            pattern = re.search(r"\w+", to_re_id)
            ini, fin = pattern.span()
            final_id = to_re_id[ini:fin]
            output["artist_ids"][i] = final_id
            if final_id not in classical:
                genres = sp.artist(f'spotify:artist:{final_id}')['genres']
                if 'classical' in genres or 'classical era' in genres:
                    classical[final_id] = 1
                else:
                    classical[final_id] = 0

        if i%100 == 0:
            t2 = process_time()
            print(f"Fila {i} --> {t2-t1} s")

    except Exception as ex:
        print(f"Ha ocurrido el siguiente error en la fila {i}\n{ex}. Para más
información, consulta el archivo traceback.txt")
        output.to_csv("/root/work/Outputs/final_data.csv", index = False)

        classic_info = ""
        for k,v in classical.items():
            classic_info += f"{k};{v}\n"
        with open("/root/work/Outputs/classical.csv", "w") as f:
            f.write(classic_info)
```

```
        error = f"{i}\n"
        error += f"Tipo de error: {sys.exc_info()[0]}\n"
        error += f"Valor del error: {sys.exc_info()[1]}\n"
        error += f"Traceback del error: {sys.exc_info()[2]}\n"

        with open("/root/work/Outputs/traceback.txt", "w") as f:
            f.write(error)
        break

    except KeyboardInterrupt:
        print(f"Has detenido el programa en la fila {i}. Para más información,
consulta el archivo traceback.txt")
        output.to_csv("/root/work/Outputs/final_data.csv", index = False)

        classic_info = ""
        for k,v in classical.items():
            classic_info += f"{k};{v}\n"
        with open("/root/work/Outputs/classical.csv", "w") as f:
            f.write(classic_info)

        error = f"{i}\n"
        error += f"Tipo de error: {sys.exc_info()[0]}\n"
        error += f"Valor del error: {sys.exc_info()[1]}\n"
        error += f"Traceback del error: {sys.exc_info()[2]}\n"
        with open("/root/work/Outputs/traceback.txt", "w") as f:
            f.write(error)
        break
print("I'm done")
```