Tendencia musical de las últimas dos décadas según datos de Spotify

Se ha realizado un Exploratory Data Analysis (EDA) para determinar cuál ha sido la tendencia musical de los últimos 20 años teniendo como punto de partida el data set Top Hits Spotify 2000-2019, adquirido de Kaggle y proporcionado por el usuario Mark Koverha.

El objetivo final del EDA es, además de entender el cambio de tendencias en el Top Spotify en los últimos años, y por ende, hacernos una idea de los gustos y modas a nivel mundial, poder sacar conclusiones sobre qué hace que una canción sea un “hit” y poder predecir futuros hit en siguientes fases del proyecto, o en vez de predecir, producir un hit.

Pasos del proyecto

1. Elegir temática
2. Obtención de datos. ¿Puedes llevar a cabo el proyecto con estos datos?
3. Define tu hipótesis. ¿Qué piensas que puedes obtener de estos datos? ¿Qué vas a poder resolver? ¿Cómo lo vas a llevar a cabo?
4. Limpia los datos: duplicados, missings, columnas inútiles...
5. Exploratorio: obtén todos los estadísticos y gráficos que necesites para entender bien tu dataset.
6. Concluye con tu análisis si estabas o no en lo cierto acerca de tu planteamiento y tu hipótesis.

Elección de la temática:

Siempre digo que la mejor inversión que he hecho es, por un lado, unos altavoces BOSE Soundlink II, y por otro, pagar mensual y religiosamente Spotify Premium.

Mi devoción por la música viene desde muy pequeño, y es por ello que tengo tres guitarras, un bajo, dos armónicas y la flauta del cole.

La elección ha sido fácil, desde hace cinco años cuando por primera vez Spotify me lanzó mis estadísticas personales. Qué géneros escuchaba más, ritmo medio, tiempo total de escuchado en todo el año… Me fascinó.

He tenido la suerte, además, de encontrar un dataset más que conveniente en Kaggle.

En un principio el proyecto iba a ser diferente, pero la falta de conocimientos, tiempo y competencias me pusieron un reto muy interesante. Sé que podría superarlo, pero no en el tiempo que tenía disponible, por lo que descarté la idea después de empezar y tirar por algo más sencillo.

La idea era, en un principio, recopilar un dataset de los Billboard Top 100 desde su primer registro en 1958, cruzarlo con las medidas que utiliza Spotify y sacar estadísticas aún más completas.

Finalmente, decidí realizar lo que me proponía, pero con el dataset del top de Spotify, para determinar el cambio de tendencia en la música occidental que escuchamos en la radio y discotecas, y poder llegar a predecir los futuros años musicales.

Así nace la idea de realizar un EDA sobre la Tendencia musical de las últimas dos décadas según datos de Spotify.

Obtención de datos:

Se ha realizado un Exploratory Data Analysis (EDA) para determinar cuál ha sido la tendencia musical de los últimos 20 años teniendo como punto de partida el data set Top Hits Spotify 2000-2019, adquirido de Kaggle y proporcionado por el usuario Mark Koverha.

Tras bajarme el dataset y comprobar que es realmente completo y sobre todo con las métricas que me interesaban (y más que he podido sacar partido como la clave y tonalidad, cosa que no me esperaba encontrar), he empezado a trabajarlo.

Definición de hipótesis:

El objetivo final del EDA es, además de entender el cambio de tendencias en el Top Spotify, y por ende, hacernos una idea de los gustos y modas a nivel mundial (al menos en occidente), poder sacar conclusiones sobre qué hace que una canción sea un “hit” y poder predecir futuros hit en siguientes fases del proyecto, o en vez de predecir, producir un hit.

Por las métricas proporcionadas en el dataset, se han realizado varias hipótesis que más tarde hemos confirmado y desmentido, como por ejemplo:

- El género más escuchado será el Pop.
- Correlación positiva entre energy y danceability.
- Correlación positiva entre tempo y energy (cuanto mayor el tempo, más energía).
- Correlación positiva entre loudness y energy (a más energía, más alto el volumen de la canción).
- Las canciones no han cambiado demasiado en cuanto a longitud. Es decir, duran más o menos lo mismo.
- Correlación entre tonalidad mayor y valence por encima de la media.

Se espera obtener también qué clave y tonalidad utilizan los géneros más populares, en los que se presupone que predominarán las claves y tonalidades Do mayor y Sol mayor (no así sus contrapartes A menor y Mi menor, respectivamente), al ser estas las más utilizadas en las canciones pop y derivados.

Carga y limpieza de datos:

Se han utilizado las siguientes librerías:

```
import pandas as pd
import numpy as np
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
```

y una vez obtenido el dataframe abriendo directamente el .CSV que nos hemos descargado:

```
spot = pd.read_csv("songs_normalize.csv")
spot.head(10)
```

Hemos obtenido lo siguiente:

	artist	song	duration_ms	explicit	year	popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	genre
0	Britney Spears	Oops!...I Did It Again	211160	False	2000	77	0.751	0.834	1	-5.444	0	0.0437	0.30000	0.000018	0.3550	0.894	95.053	pop
1	blink-182	All The Small Things	167066	False	1999	79	0.434	0.897	0	-4.918	1	0.0488	0.01030	0.000000	0.6120	0.684	148.726	rock, pop
2	Faith Hill	Breathe	250546	False	1999	66	0.529	0.496	7	-9.007	1	0.0290	0.17300	0.000000	0.2510	0.278	136.859	pop, country
3	Bon Jovi	It's My Life	224493	False	2000	78	0.551	0.913	0	-4.063	0	0.0466	0.02630	0.000013	0.3470	0.544	119.992	rock, metal
4	*NSYNC	Bye Bye Bye	200560	False	2000	65	0.614	0.928	8	-4.806	0	0.0516	0.04080	0.001040	0.0845	0.879	172.656	pop
5	Sisqo	Thong Song	253733	True	1999	69	0.706	0.888	2	-6.959	1	0.0654	0.11900	0.000096	0.0700	0.714	121.549	hip hop, pop, R&B

Un total de 2000 filas y 18 columnas:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 18 columns):
```

No todas nos valen, por lo que nos deshacemos de algunas como “explicit” y “popularity”, pues no vamos a sacar ningún tipo de datos que nos interese con estos parámetros. En realidad, podríamos deshacernos también de las columnas “artist” y “song”, sin embargo las dejamos por tener una referencia.

```
#Quitamos columnas que no necesitamos:

spot.drop(["explicit", "popularity"], axis=1)
spot.head()
```

Como podemos observar, en el dataset tenemos canciones lanzadas en 1999. Las vamos a descartar también, por querer analizar las canciones del siglo XXI, por muy populares que sean las anteriores durante los primeros años de siglo.

A este nuevo dataframe lo llamamos spot00 (de los 2000s).

```
#Limpiamos el df, queremos canciones lanzadas entre el 2000 y 2020,
algunas nos sobran
#Usamos un masking
#
https://www.interviewqs.com/ddi-code-snippets/select-pandas-dataframe-r
ows-between-two-dates

start_date = 1999
end_date = 2021

mask = (spot["year"] > start_date) & (spot["year"] <= end_date)
spot00 = spot.loc[mask]
spot00.head()
```

También observamos en el dataframe original que tanto “key” (clave en la que está la canción) y “mode” (la tonalidad de la misma), están en valores numéricos, del 0 al 11 para key y 1 y 0 para mode, respectivamente.

Sabemos que las claves musicales van de Do a Si en semitonos, aunque el cambio de Mi a Fa y de Si a Do es un tono completo. Es decir.

Do, Do#/Reb, Re, Re#/Mib, Mi, Fa, Fa#/Solb, Sol, Sol#/Lab, La, La#/Sib, Si, Do, …

Para tener todo más claro mientras trabajamos el dataframe, cambiaremos de nombre tanto los valores de Key como los de Mode. Por la documentación del dataset, sabemos que 1=Mayor, 0=Menor y que la escala empieza por Do, es decir, 0=Do, 1=Do#, 2=Re, y así sucesivamente:

```
#Cambiar numero claves por escala musical Do Re Mi etc
```

```
# https://datatofish.com/replace-values-pandas-dataframe/
#df["column name"] = df["column name"].replace({1st old value:"1st new value",2nd old value:"2nd new value",...}, inplace=True)

spot00["key"].replace({0:"Do",
1:"Do#",
2:"Re",
3:"Re#",
4:"Mi",
5:"Fa",
6:"Fa#",
7:"Sol",
8:"Sol#",
9:"La",
10:"La#",
11:"Si"}, inplace=True)


#Cambiar numero tonalidad por mayor y menor
# https://datatofish.com/replace-values-pandas-dataframe/
#df["column name"] = df["column name"].replace({1st old value:"1st new value",2nd old value:"2nd new value",...}, inplace=True)

spot00["mode"].replace({0:"Menor",
1:"Mayor"}, inplace=True)
spot00.head()
```

Con el Dataset a nuestro gusto, empezamos con el análisis.

Exploratorio:

Línea temporal:

Uno de los puntos más interesantes a descubrir es cuánto dura la canción hoy en día y cuánto duraban antes. A priori no se esperaba un cambio notorio en ello.
Es interesante averiguar esto porque, de cara a un futuro avance en el proyecto que nos pueda ayudar a predecir un hit (o producirlo), saber por donde movernos.

```
# Media tiempo por año, de menos a más tiempo
spot00.groupby(["year"])["duration_ms"].mean().round().sort_values(ascending=False)
```

Metemos esto en un nuevo dataframe llamado anyos_ms (será lo que dura la canción de media cada año en milisegundos), para poder trabajar cómodamente:

```
#Nuevo df
```

```
anyos_ms =
pd.DataFrame(spot00.groupby(["year"])["duration_ms"].mean().round())
anyos_ms
anyos_ms.transpose()
```

Obtenemos el siguiente resultado:

year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	...
duration_ms	249993.0	242450.0	253549.0	236335.0	234040.0	236514.0	238148.0	231750.0	235675.0	236082.0	...

1 rows × 21 columns

Preferimos tenerlo en una nomenclatura a la que estamos más acostumbrados, es decir, min:seg.

```
# pasar el tiempo a minutos:segundos
#https://stackoverflow.com/questions/46929396/convert-milliseconds-colu
mn-to-hhmmssms-datetime-in-pandas

# spot00["duration_ms"].dtypes #sabemos que duration_ms es int
anyos_mins =
pd.DataFrame(spot00.groupby(["year"])["duration_ms"].mean().round())
anyos_mins

anyos_mins["duration_ms"] = pd.to_datetime(anyos_mins["duration_ms"],
unit="ms").dt.strftime("%M:%S.%f").str[:-7]
# spot00.dtypes
anyos_mins.dtypes
anyos_mins

anyos_mins.transpose()
```

Et voilà:

year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	...
duration_ms	04:09	04:02	04:13	03:56	03:54	03:56	03:58	03:51	03:55	03:56	...

Veamos cuales es la media anual más baja, la más alta, y media global en 20 años:

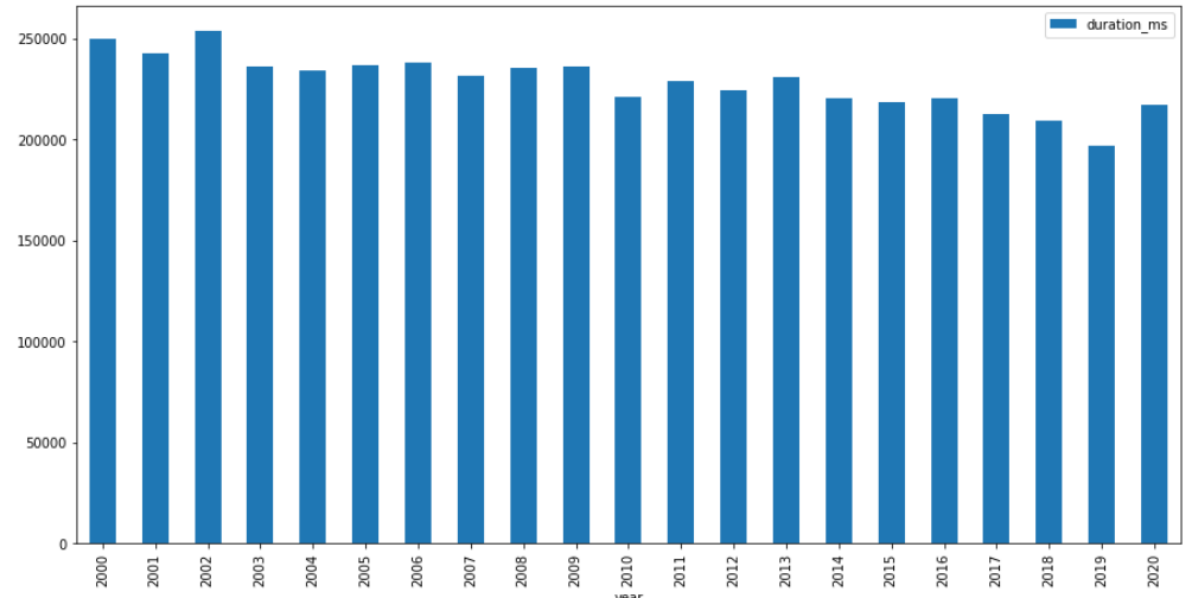
```
minimo=anyos_mins.min()
maximo=anyos_mins.max()

sec=anyos_ms.mean()/1000
min=sec//60
seg=sec%60
```

```
media = "%d:%d" % (min,seg)

print("Canción que menos dura{}, que más dura{} y media{}"
.format(minimo, maximo, media))
```

```
Canción que menos duraduration_ms    03:16
dtype: object, que más duraduration_ms    04:13
dtype: object y media3:48
```

Y veamos cómo se distribuye a lo largo de los años mediante una gráfica:

```
#Gráfico duración años
anyos_ms.plot.bar(figsize=(15,7.5))
```



Con esta gráfica delante, podemos ver con claridad cómo la duración media de canción ha ido descendiendo en las últimas décadas, llegando a su valor mínimo en 2019 (3:16), y siendo su máximo en 2002 (4:13).

La tendencia ha sido decreciente, reduciendo hasta casi un minuto si comparamos los valores más altos (inicios del 2000) con su mínimo (finales de los 2010). Podemos decir que las canciones son más cortas desde el año 2000, teniendo una clara diferenciación a partir del 2010.

Para ver esto de manera más clara, realizamos la media de la primera y segunda decada para compararlas:

```
print(anyos_ms.iloc[0:10].mean().round())
print(anyos_ms.iloc[11:20].mean().round())
```

```
duration_ms    239454.0
dtype: float64
duration_ms    217893.0
dtype: float64
```

Vemos de manera clara que en la segunda década (2010-2019) las canciones duran menos.

Clave y tonalidad:

Como se ha mencionado, un dato interesante que no sabíamos que tendríamos es la tonalidad y clave de la canción.

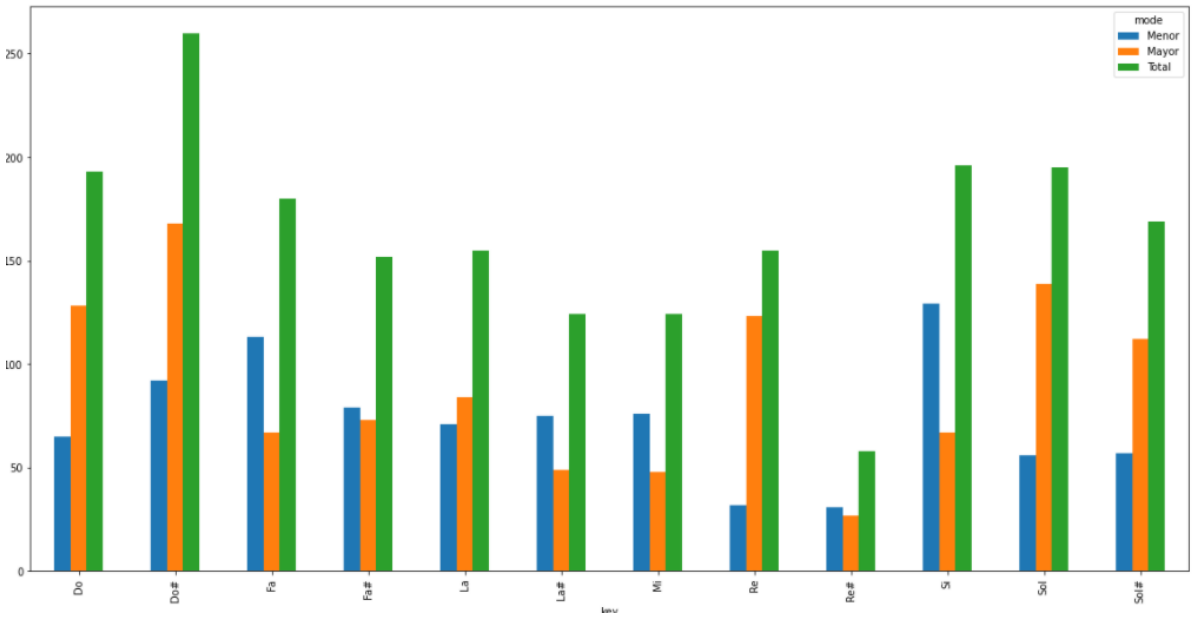
Para analizar la tonalidad de la canción, vamos a separar la clave y la tonalidad en un nuevo DataFrame que incluya solo estos dos parámetros, separados por tonalidad (dos columnas), clave, y el total.

```
clave_tono_mm =
spot00.groupby(["key","mode"]).size().unstack(fill_value =
0).reset_index()


clave_tono_mm ["Total"]= clave_tono_mm ["Mayor"]+clave_tono_mm
["Menor"]

clave_tono_mm
```

mode	key	Mayor	Menor	Total
0	Do	128	65	193
1	Do#	168	92	260
2	Fa	67	113	180
3	Fa#	73	79	152
4	La	84	71	155
5	La#	49	75	124
6	Mi	48	76	124
7	Re	123	32	155
8	Re#	27	31	58
9	Si	67	129	196
10	Sol	139	56	195
11	Sol#	112	57	169

Así, obtenemos la cantidad total de canciones que hay por clave y tonalidad, además.

Con estos datos podremos sacar diferentes gráficas representativas para entender todo de manera más visual:



Podemos ver que la inmensa mayoría de las canciones están en la tonalidad de DO#, muy parecido pero diferente a nuestra predicción (que sería Do). Tiene lógica.

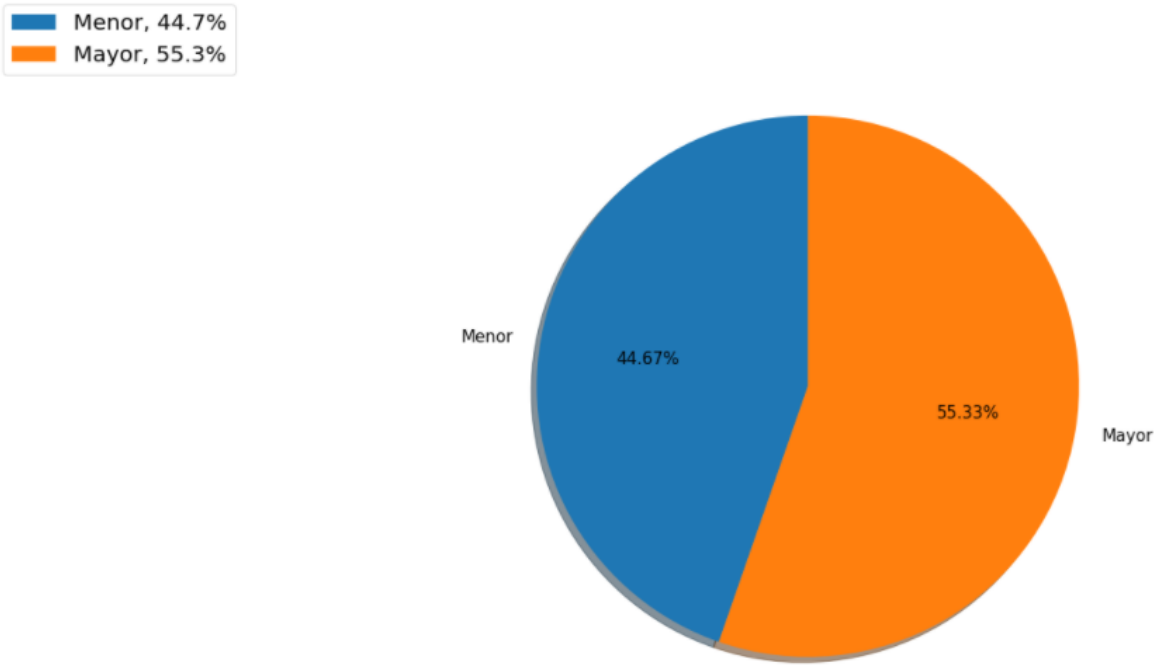
Podemos intuir que la tonalidad predominante en general es la tonalidad mayor, y en este caso en concreto, la mayoría de las canciones están compuestas en Do# mayor.

Vamos a confirmarlo:

```
may_tot=clave_tono_mm.Mayor.sum()
print(may_tot)
min_tot=clave_tono_mm.Menor.sum()
print(min_tot)
```

```
1085
876
```

```
labels= ["Menor","Mayor"]
sizes = (min_tot, may_tot)

fig1, ax1 = plt.subplots( figsize=(20,10))
ax1.pie(sizes, labels=labels, textprops={'fontsize': 15},
autopct='%1.2f%%',
        shadow=True, startangle=90)
#esto es para poner la leyenda
plt.legend(
    loc='upper left',
    labels=['%s, %1.1f%%' % (
        l, (float(s) / (sum(sizes))) * 100) for l, s in zip(labels,
sizes)],
    prop={'size': 20},
    bbox_to_anchor=(0.0, 1),
    bbox_transform=fig1.transFigure
)

ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a
circle.
```



Efectivamente, la mayoría de las canciones están en tonalidad MAYOR (en un 55.33% del total).

En general, las siguientes claves y tonalidades más utilizadas son Si (196 canciones, donde predomina la tonalidad menor), Sol (195) y Do (193) (ya habíamos previsto Do y Sol como las claves más utilizadas), donde sorprende la irrupción de la clave en Si.

Se dice que una tonalidad MENOR está relacionada con canciones más tristes, mientras que la tonalidad MAYOR se relaciona con canciones más alegres. Veremos si podemos confirmar esta correlación.

Géneros más escuchados:

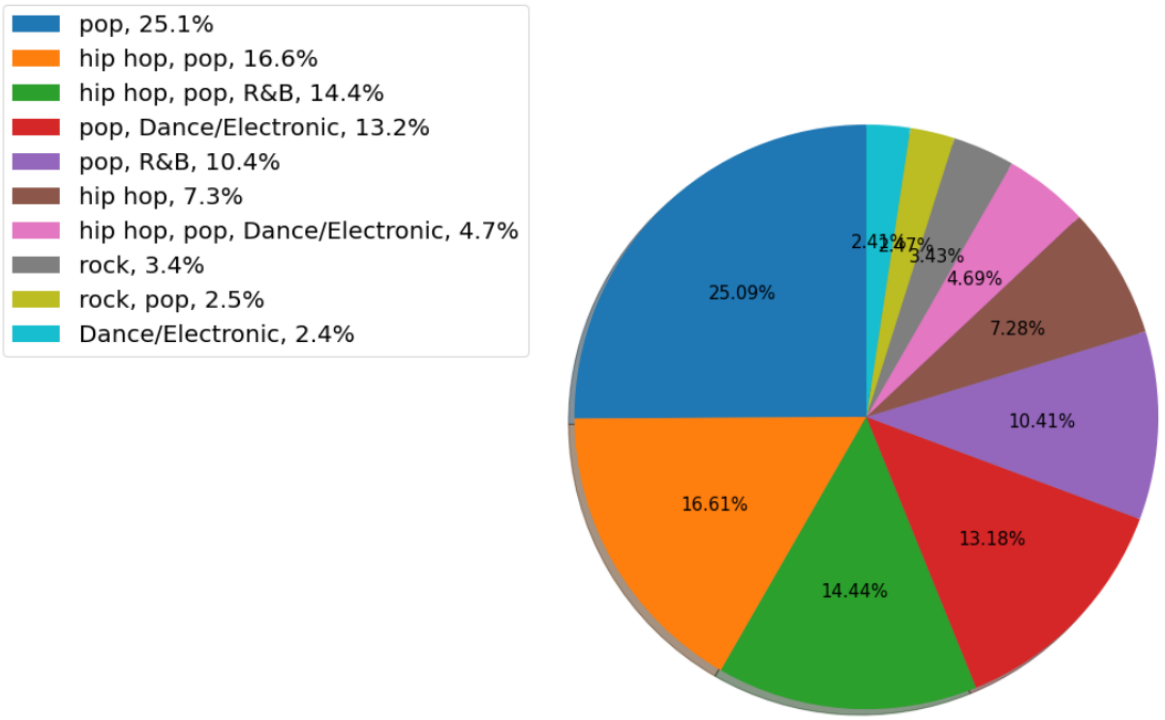
En total, hemos obtenido 57 géneros diferentes en el dataset de Spotify. Cabe recordar que, en muchas ocasiones Spotify tiende a mezclar géneros entre sí. Por ejemplo, tenemos por un lado el género pop, el hip-hop por otro y Dance por otro.
A la vez, tenemos el genero “pop, hip-hop” o “pop, dance/electronic”.
Al tener tantos géneros dentro de los top hit de Spotify, vamos a descubrir cuáles son los top 10 géneros.

```
generos =
pd.DataFrame(spot00["genre"].value_counts().rename_axis("genero").reset
_index(name="Total"))
generos.head(10).sort_values(by="Total", ascending=False)
```

	genero	Total
0	pop	417
1	hip hop, pop	276
2	hip hop, pop, R&B	240
3	pop, Dance/Electronic	219
4	pop, R&B	173
5	hip hop	121
6	hip hop, pop, Dance/Electronic	78
7	rock	57
8	rock, pop	41
9	Dance/Electronic	40

Vemos que predomina, por mucho, el género pop, algo nada sorprendente, con un total de 417 canciones (sin tener en cuenta sus primos-hermanos), acaparando más del 25% del top 10.

```
labels= generos["genero"].head(10)
sizes = generos["Total"].head(10)

fig1, ax1 = plt.subplots( figsize=(20,10))
ax1.pie(sizes, textprops={'fontsize': 15}, autopct='%1.2f%%', #añadir
labels=labels detrás de sizez, para que aparezcan en el gráfico los
nombres
        shadow=True, startangle=90)
#esto es para poner la leyenda
plt.legend(
    loc='upper left',
    labels=['%s, %1.1f%%' % (
        l, (float(s) / (sum(sizes))) * 100) for l, s in zip(labels,
sizes)],
    prop={'size': 20},
    bbox_to_anchor=(0.0, 1),
    bbox_transform=fig1.transFigure
)

ax1.axis('equal')
```



Métricas Spotify:

Spotify cuenta con ciertas métricas propias que otorga a cada canción. Las métricas representan las características de la canción en cuestión. Estas métricas son:

Danceability, energy, loudness, speechiness, acousticness instrumentalness, liveness y valence.

Nos interesa analizar estas métricas por tres razones. La primera es ver la correlación que existe entre ellas, la correlación con los géneros y la correlación con los demás datos del dataset.

Veamos la correlación entre ellas en total:

```
f = plt.figure(figsize=(10, 10))
corr_tot = medidas.corr()

corr_tot.style.background_gradient(cmap='coolwarm')
```

	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence
danceability	1.000000	-0.108289	-0.036767	0.144287	-0.066097	0.022460	-0.123235	0.399328
energy	-0.108289	1.000000	0.654141	-0.058530	-0.444048	0.035678	0.153071	0.332772
loudness	-0.036767	0.654141	1.000000	-0.082725	-0.309823	-0.103736	0.104099	0.236078
speechiness	0.144287	-0.058530	-0.082725	1.000000	0.001260	-0.063182	0.059650	0.075088
acousticness	-0.066097	-0.444048	-0.309823	0.001260	1.000000	-0.004326	-0.109464	-0.128916
instrumentalness	0.022460	0.035678	-0.103736	-0.063182	-0.004326	1.000000	-0.040371	-0.018708
liveness	-0.123235	0.153071	0.104099	0.059650	-0.109464	-0.040371	1.000000	0.018475
valence	0.399328	0.332772	0.236078	0.075088	-0.128916	-0.018708	0.018475	1.000000

En el cómputo global de canciones del Top Hit Spotify 2000-2020, podemos ver que existe una fuerte correlación entre la energía de la canción y el volumen de la misma.
Así mismo, se observa también una correlación entre su positividad y energía y su positividad e índice de danceability.

Al igual, se ve una correlación negativa entre las canciones con alta acousticness y su energía, denotando que las canciones acústicas no son muy enérgicas por lo general.

Si aplicamos esto al top 10, nos quedamos con esta gráfica:

```
f = plt.figure(figsize=(10, 10))
corr_top = medgen_top10.corr()

corr_top.style.background_gradient(cmap='coolwarm')
```

	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence
danceability	1.000000	-0.559483	-0.393690	0.724511	0.425983	-0.176803	-0.278883	0.596272
energy	-0.559483	1.000000	0.762230	-0.613102	-0.809893	0.537115	0.543229	-0.488801
loudness	-0.393690	0.762230	1.000000	-0.315056	-0.437317	-0.026130	0.506879	-0.094330
speechiness	0.724511	-0.613102	-0.315056	1.000000	0.359582	-0.472338	0.115067	0.435873
acousticness	0.425983	-0.809893	-0.437317	0.359582	1.000000	-0.606204	-0.569775	0.407468
instrumentalness	-0.176803	0.537115	-0.026130	-0.472338	-0.606204	1.000000	-0.043396	-0.555558
liveness	-0.278883	0.543229	0.506879	0.115067	-0.569775	-0.043396	1.000000	-0.420346
valence	0.596272	-0.488801	-0.094330	0.435873	0.407468	-0.555558	-0.420346	1.000000

En el top 10 podemos ver que existen correlaciones (tanto positivas como negativas) más marcadas, y algunas totalmente diferentes.

Confirmamos que en el top 10 también hay una (aún más fuerte) correlación entre energy y loudness.

Algo sorprendente es la correlación positiva que existe entre danceability y speechiness. A priori, podría parecer un error, sin embargo, con el auge de los últimos años de géneros como el trap o artistas del New School Rap, podría entenderse. Es algo que no podemos confirmar y habría que hacer un análisis más exhaustivo, como ver qué artistas son los más populares en los últimos años y a qué género pertenecen.

A nivel global, teniendo en cuenta todas las variables del set la tabla nos quedaría así:

```
f = plt.figure(figsize=(10, 10))
corr_top = spot.corr()

corr_top.style.background_gradient(cmap='coolwarm')
```

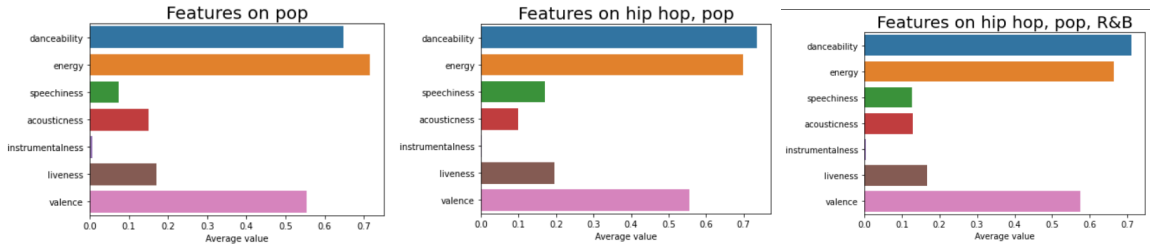
	duration_ms	explicit	year	popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo
duration_ms	1.000000	0.123595	-0.316534	0.050617	-0.060057	-0.078763	-0.002560	-0.079912	-0.003848	0.066998	0.010923	-0.004208	0.024941	-0.116870	-0.028603
explicit	0.123595	1.000000	0.078477	0.046605	0.248845	-0.162462	0.003320	-0.089829	0.049576	0.417343	-0.033523	-0.082522	0.008884	-0.045455	0.013221
year	-0.316534	0.078477	1.000000	-0.003825	0.033532	-0.108644	0.007380	0.017479	-0.007358	0.001111	0.033809	-0.050265	-0.027037	-0.209365	0.076867
popularity	0.050617	0.046605	-0.003825	1.000000	-0.003546	-0.014021	0.014823	0.030632	-0.021353	0.021162	0.024619	-0.048059	-0.009856	-0.016142	0.014288
danceability	-0.060057	0.248845	0.033532	-0.003546	1.000000	-0.104038	0.032731	-0.033315	-0.067528	0.145590	-0.065429	0.023207	-0.126413	0.403178	-0.173418
energy	-0.078763	-0.162462	-0.108644	-0.014021	-0.104038	1.000000	-0.003446	0.651016	-0.040651	-0.057018	-0.445469	0.037861	0.156761	0.334474	0.153719
key	-0.002560	0.003320	0.007380	0.014823	0.032731	-0.003446	1.000000	-0.007474	-0.153182	0.007147	0.002365	-0.008173	-0.033071	0.036977	-0.001431
loudness	-0.079912	-0.089829	0.017479	0.030632	-0.033315	0.651016	-0.007474	1.000000	-0.028133	-0.076388	-0.310039	-0.104925	0.102159	0.232150	0.080709
mode	-0.003848	0.049576	-0.007358	-0.021353	-0.067528	-0.040651	-0.153182	-0.028133	1.000000	-0.000077	0.005744	-0.038613	0.025439	-0.074681	0.048434
speechiness	0.066998	0.417343	0.001111	0.021162	0.145590	-0.057018	0.007147	-0.076388	-0.000077	1.000000	0.000394	-0.062954	0.061172	0.073605	0.057747
acousticness	0.010923	-0.033523	0.033809	0.024619	-0.065429	-0.445469	0.002365	-0.310039	0.005744	0.000394	1.000000	-0.005214	-0.110043	-0.128128	-0.103660
instrumentalness	-0.004208	-0.082522	-0.050265	-0.048059	0.023207	0.037861	-0.008173	-0.104925	-0.038613	-0.062954	-0.005214	1.000000	-0.034897	-0.015192	0.034608
liveness	0.024941	0.008884	-0.027037	-0.009856	-0.126413	0.156761	-0.033071	0.102159	0.025439	0.061172	-0.110043	-0.034897	1.000000	0.019040	0.028636
valence	-0.116870	-0.045455	-0.209365	-0.016142	0.403178	0.334474	0.036977	0.232150	-0.074681	0.073605	-0.128128	-0.015192	0.019040	1.000000	-0.025076
tempo	-0.028603	0.013221	0.076867	0.014288	-0.173418	0.153719	-0.001431	0.080709	0.048434	0.057747	-0.103660	0.034608	0.028636	-0.025076	1.000000

Los top 10 géneros según sus métricas:

Tenemos por un lado los géneros más escuchados, y por otro lado tenemos las características principales que otorga Spotify a cada canción, viendo las correlaciones que tienen entre sí.

Es hora de juntar ambas, y ver qué características predominan en cada género musical del top 10.

```
medgen_top10

for genre in gen_validos:
    data = medgen_top10[medgen_top10.genre == genre]
    #Exclude features with different scales
    data = data.drop(["genre", "loudness"],
axis=1).transpose().reset_index()
    data.columns = ["feature", "value"]
    plt.clf()
    ax = sns.barplot(x="value", y="feature", data=data)
    ax.set_xlabel("Average value")
    ax.set_ylabel("")
    ax.set_title("Features on " + genre, fontsize=20)
    plt.show()
```



(Por motivos de espacio, dejaré solo el top 3, sin embargo las conclusiones se sacan a raíz del top 10)

Podemos ver con claridad que dentro del top 10 géneros, existe una verdadera predominancia por lo bailable que es la canción, la energía que transmite y que en bastante good vibes.

Conclusiones:

Tras realizar el EDA, hemos conseguido por un lado afirmar hipótesis que habíamos hecho, nos hemos encontrado también con varias sorpresas y en general, podemos afirmar que:

- Existe una fuerte correlación entre la energía de la canción con su volumen.
- Existe correlación notable entre la energía de la canción y su danceability, lo bailable que es con el positivismo que desprende, y la energía y su positivismo.
- La gran mayoría de canciones está compuesta en clave de Do#, y en la mayoría de los casos es en tonalidad mayor.
- Hay una tendencia descendente en lo que a durabilidad de la canción se refiere.
- El género más escuchado es el pop, seguido de “primos-hermanos”.

Para finalizar, podríamos decir que en los últimos años, un hit ha sido una canción pop en clave Do# y tonalidad mayor, con mucha energía, muy bailable y que transmite good vibes.

O que la manera más fácil de componer un hit es siguiendo estas directrices.

La mayor dificultad ha sido que mi cabeza quería conseguir mucho, pero no sabía cómo hacerlo. En este aspecto, se ha empleado muchísimo tiempo en prueba/error y documentarse en internet. Ha llegado un momento en el que he tenido que copiar-pegar un código para poder avanzar, pero en un alto porcentaje lo he hecho todo yo.

Es como de verdad me he dado cuenta que tengo que practicar más, y a la par, la manera en la que más he aprendido y avanzado.

El dataset contiene las siguientes columnas, las cuales, por conveniencia y mayor comprensión de los datos, vamos a explicar, pues se utiliza nomenclatura musical con la cual no todo el mundo está familiarizado:

- artist: Nombre del artista.
- song: Título de la canción.
- duration_ms: cuánto dura la canción, en milisegundos.
- explicit: canciones cuyo contenido podría considerar explícito por sus letras.
- year: Año debut canción.
- popularity: Popularidad, del 0 al 1. Cuanto más alto, más popular.
- danceability: como de apropiada es la canción para bailar, basado en diferentes parámetros como el tempo, estabilidad de ritmo, fuerza del beat, etc. Escala del 0 al 1, cuanto más alto, más bailable es la canción.
- energy: Representa la intensidad de la canción.
- key: la clave en la que está la canción. Do, Re, Mi, Fa…
- loudness: decibelios medios de la canción.
- mode: tonalidad. Indica si la canción está en modalidad mayor o menor. Re menor, Do mayor, etc. Mayor se representa con 1, menor con 0.
- speechiness: Indica la presencia de palabras habladas (no cantadas). Cuanto más se hable (recitar un poema, audiolibro, etc.) más cerca del 1 está. Valores entre el 0.33 y 0.66 indica canciones que tienen palabras cantadas y habladas, como es el caso del rap o hip-hop..
- acousticness: probabilidad de que la canción sea acústica.
- instrumentalness: predice si la canción tiene vocales. Sonidos como “ooh”, “aah”s, se consideran instrumentales. Cuanto más cerca del 1, menos probable es que la canción contenga vocales.
- liveness: detecta si hay presencia de público en la grabación, es decir si es una versión de la canción grabada en directo, por ejemplo.
- valence: describe la positividad de la canción. Normalmente, una canción en tonalidad mayor se considera una canción “alegre”, por lo que se espera cierta correlación. Cuanto más alto el valence, más positividad (felicidad, euforia, etc), cuanto más bajo, más tristeza, enfado, etc. denota la canción.
- tempo: tempo estimado de la canción, en Beat Per Minutes. El tempo es la velocidad o ritmo de la canción.
- genre: género de la canción.

https://findyourmelody.com/major-vs-minor-scales-differences/

https://mixedinkey.com/captain-plugins/wiki/common-chord-progressions-pop-music/#:~:text=C%20major%20and%20G%20major,and%20scales%20for%20Pop%20music.