Data Quality Engineer – Prueba Técnica

Reto: Spotify API

Reporte de calidad de datos

R5

Oscar Mario Mariño Arias

05/01/2024

Tabla de contenido

Tabla de figuras

Tabla de tablas

Data Quality Engineer – Prueba Técnica
Reto: Spotify API

Reporte de Calidad de datos

R5

1. Introducción:

A partir de un archivo proveniente de la API de Spotify, se realizó un análisis exhaustivo de datos enfocado en encontrar las anomalías del dataset, para posteriormente realizar una evaluación y categorización de estos. Dicho trabajo fue llevado a cabo haciendo uso de Python y la librería Pandas.

2. Metodología:

Para realizar la evaluación de la calidad de los datos se adoptó el enfoque mencionado en DAMA UK (2018):

1. Identificar los datos que deben ser evaluados en calidad.
2. Definir las dimensiones de calidad a emplear y el peso de estas para la evaluación de los datos.
3. Definir los valores y rangos que representan buena y mala calidad de los datos para las diferentes dimensiones.
4. Aplicar los criterios de evaluación al set de datos.
5. Analizar y definir si la calidad de los datos es aceptable.
6. Cuando sea el caso, adoptar medidas correctivas para mejorar la calidad de los datos y evitar futuras recurrencias.

3. Dimensiones de calidad de datos:

A continuación, se describen las dimensiones utilizadas para la evaluación de la calidad de los datos.

Completitud (Completeness): Hace referencia a la proporción de los datos almacenados con relación a su totalidad. Medida de ausencia de valores sin información (nulos).

Unicidad (Uniqueness): Dimensión que hace alusión a si los registros en el set de datos son de carácter único.

Validez (Validity): Los datos son válidos si se ajustan a la sintaxis de su definición, tanto en el formato, como el tipo y rango.

Precisión (Accuracy): Hace referencia al grado en el cual los datos representa correctamente el evento que están describiendo.

Consistencia (Consistency): Es la ausencia de diferencias en la comparación de dos o más registros con relación a la definición del campo.

Es importante mencionar que de acuerdo con DAMA UK (2018), las diferentes dimensiones utilizadas para la evaluación de datos pueden tener diferentes pesos, lo que a su vez repercute en la medida de calidad obtenida para los datos. Para beneficio del siguiente reporte, las dimensiones serán tomadas con igual

peso a la hora de evaluar la información suministrada. Así mismo, todas las características del archivo "dataset.csv" serán analizadas.

4. Rangos de evaluación:

Para la evaluación de los datos y definición de su calidad se fijaron los siguientes rangos:

Tabla 1. Rangos de e

Rango	Calidad
100 – 96	Buena
<96	Aceptable
<90	Regular
<85	Mala

Dependiendo de la cantidad de errores los datos serán definidos dentro de un rango, lo cual dará como resultado la calidad de estos.

5. Anomalías:

A partir del análisis del set de datos suministrado, el cual cuenta con un total de 539 registros (filas) y 27 características (columnas), múltiples errores fueron evidenciados. Estos fueron descritos con relación a las dimensiones de calidad y son presentados a continuación:

3.1. Completitud

Características como *track_id*, *track_name*, *danceability*, *energy*, *key*, *loudness*, *speechiness*, *liveness*, *time_signature* y *album_name* presentan registros faltantes (Figura 1). La Tabla 2 detalla la cantidad de errores encontrados para las diferentes características en la dimensión completitud.

```
# Verificamos la completitud de los datos.
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 539 entries, 0 to 538
Data columns (total 27 columns):
 #   Column                        Non-Null Count
---  ------                        --------------
 0   disc_number                   539 non-null
 1   duration_ms                   539 non-null
 2   explicit                      539 non-null
 3   track_number                  539 non-null
 4   track_popularity              539 non-null
 5   track_id                      535 non-null
 6   track_name                    536 non-null
 7   audio_features.danceability   538 non-null
 8   audio_features.energy         537 non-null
 9   audio_features.key            538 non-null
 10  audio_features.loudness       537 non-null
 11  audio_features.mode           539 non-null
 12  audio_features.speechiness    538 non-null
 13  audio_features.acousticness   539 non-null
 14  audio_features.instrumentalness 539 non-null
 15  audio_features.liveness       538 non-null
 16  audio_features.valence        539 non-null
 17  audio_features.tempo          539 non-null
 18  audio_features.id             539 non-null
 19  audio_features.time_signature 538 non-null
 20  artist_id                     539 non-null
 21  artist_name                   539 non-null
 22  artist_popularity             539 non-null
 23  album_id                      539 non-null
 24  album_name                    493 non-null
```

Figura 1. Completitud de los datos.

Tabla 2. Anomalías encontradas para la dimensión completitud.

Característica	No. de anomalías
track_id	4
track_name	3
**danceability*	1
**energy*	2
**key*	1
**loudness*	2
**speechiness*	1
**liveness*	1
**time_signature*	1
album_name	46

Las características marcadas con un (*) presentan los siguientes caracteres “audio_features.” previo al nombre del campo.

Las columnas del tipo flotante (float) y entero (int), presentan registros con datos NaN y None, que hacen referencia a datos faltantes o vacíos (Figura 2A). Para las columnas del tipo carácter (str), los registros sin datos son representados por valores NoneType (Figura 2B).

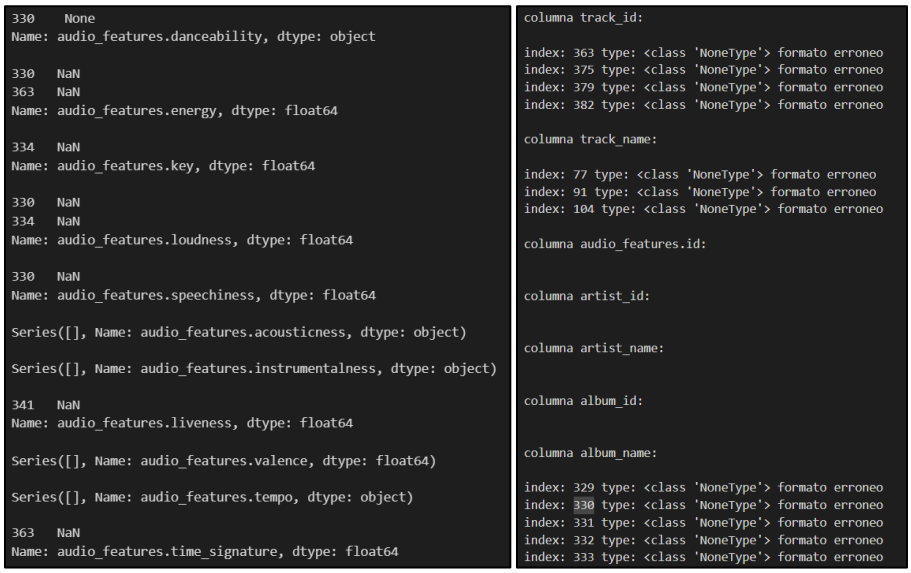

Figura 2. Registros vacíos. A) Columnas del tipo float e Int. B) Columnas del tipo str.

En la característica *album_name* es necesario tener en cuenta que, de acuerdo con la documentación de Spotify, el valor de dichos registros puede estar vacío (nulo) debido a que el álbum fue eliminado, no obstante, fue posible evidenciar que el registro existe para los álbumes “Speak Now World Tour”, y “reputation Stadium Tour Surprise Song “ en la API, por lo tanto, constituye una anomalía.

3.2. Unicidad

El set de datos presenta 19 registros que se encuentran repetidos, el álbum “Lover” con 18 registros y un registro del álbum “Midnights (The Til Dawn Edition)” (Figura 3). Debido a que en la característica "explicit" hay un error en el formato de la casilla, el registro completo es considerado como diferente, sin embargo,

al corregir el formato es posible evidenciar que la fila es un registro adicional repetido. En la Tabla 3 es posible observar la cantidad de errores relacionados con la dimensión Unicidad.

```python
# Dimensiones iniciales del set de datos.
print(f'Dimensiones iniciales: {df.shape}')
print(f'Dataset sin registros repetidos: {df.drop_duplicates().shape}. No tiene en cuenta la fila que tiene el registro con formato erroneo')
print(f'Registros repetidos: {df[df.duplicated() == True].shape}. No tiene en cuenta la fila que tiene el registro con formato erroneo')
print('')
print('Ejemplo de algunos registros repetidos:')
display(df[df.duplicated() == True][['artist_name', 'album_name','track_name']].sort_values(by= 'track_name').sample(10))

# Dado a que en la característica "explicit" hay un error en el formato de la casilla el registro es considerado como diferente, sin embargo,
# al corregir esto podemos evidenciar que es un registro adicional repetido.
df[(df['album_name'] == 'Lover') &  (df['track_name'] == 'Cruel Summer')][['explicit', 'album_name', 'track_name']].sort_values(by= 'track_name')
```

✓ 0.0s

```
Dimensiones iniciales: (539, 27)
Dataset sin registros repetidos: (521, 27). No tiene en cuenta la fila que tiene el registro con formato erroneo
Registros repetidos: (18, 27). No tiene en cuenta la fila que tiene el registro con formato erroneo

Ejemplo de algunos registros repetidos:
```

	artist_name	album_name	track_name
295	Taylor Swift	Lover	I Forgot That You Existed
306	Taylor Swift	Lover	Soon You'll Get Better (feat. The Chicks)
297	Taylor Swift	Lover	Lover
304	Taylor Swift	Lover	Death By A Thousand Cuts
303	Taylor Swift	Lover	Cornelia Street
312	Taylor Swift	Lover	Daylight
300	Taylor Swift	Lover	I Think He Knows
310	Taylor Swift	Lover	ME! (feat. Brendon Urie of Panic! At The Disco)
309	Taylor Swift	Lover	Afterglow
305	Taylor Swift	Lover	London Boy

	explicit	album_name	track_name
278	False	Lover	Cruel Summer
296	No	Lover	Cruel Summer

Figura 3**.** Registros repetidos en el set de datos.

Tabla 3. Anomalías encontradas para la dimensión unicidad.

Característica	No. de anomalías	Característica	No. de anomalías
disc_number	19	**instrumentalness*	19
duration_ms	19	**liveness*	19
explicit	19	**valence*	19
track_number	19	**tempo*	19
track_popularity	19	**id*	539
track_id	19	**time_signature*	19
track_name	19	*artist_id*	19
**danceability*	19	*artist_name*	19
**energy*	19	*artist_popularity*	19
**key*	19	*album_id*	19
**loudness*	19	*album_name*	19
**mode*	19	*album_release_date*	19
**speechiness*	19	*álbum_total_tracks*	19
**acousticness*	19	-	-

Las características marcadas con un (*) presentan los siguientes caracteres “audio_features.” previo al nombre del campo.

Además, las columnas *track_id* y *audio_features.id* pertenecientes a los endpoits Track y Track's Audio Features, hacen referencia a la misma información, por lo tanto, son columnas con información repetida (Figura 4). No obstante, dado que *track_id* tiene 4 casillas vacías NoneType y 4 casillas del tipo carácter pero sin caracteres (evidenciado más adelante), algunos registros pueden ser tomados como diferentes.

```
# Tanto la caracteristica track_id como audio_features.id (endpoits Track y Track's Audio Features), hacen referencia a la misma información.
df[['track_id', 'audio_features.id']]
```

✓ 0.0s

	track_id	audio_features.id
0	4WUepByoeqcedHoYhSNHRt	4WUepByoeqcedHoYhSNHRt
1	0108kcWLnn2HIH2kedi1gn	0108kcWLnn2HIH2kedi1gn
2	3Vpk1hfMAQme8VJ0SNRSkd	3Vpk1hfMAQme8VJ0SNRSkd
3	1OcSfkeCg9hRC2sFKB4IMJ	1OcSfkeCg9hRC2sFKB4IMJ
4	2k0ZEeAqzvYMcx9Qt5aCIQ	2k0ZEeAqzvYMcx9Qt5aCIQ
...	...	...
534	1j6gmK6u4WNI33IMZ8dC1s	1j6gmK6u4WNI33IMZ8dC1s
535	7CzxXgQXurKZCyHz9ufbo1	7CzxXgQXurKZCyHz9ufbo1
536	1k3PzDNjg38cWqOvL4M9vq	1k3PzDNjg38cWqOvL4M9vq
537	0YgHuReCSPwTXYny7isLja	0YgHuReCSPwTXYny7isLja
538	1hxLyjC9D9Jpw6EAPKqWv4	1hxLyjC9D9Jpw6EAPKqWv4

539 rows × 2 columns

Figura 4. Características con el mismo tipo de información.

Debido a que las diferentes dimensiones se encuentran correlacionadas, la presencia de datos repetidos genera que la columna con el número de canciones por álbum no sea acorde con el número real de registros por álbum (número de canciones), como será evidenciado más adelante.

3.3. Validez

Como se evidencio anteriormente, las diferentes dimensiones se encuentran relacionadas e influyen entre sí. En este caso, la validez de los datos es afectada por la completitud de los registros.

La columna *explicit* del tipo booleano presenta 5 registros de cadena de caracteres que son errores para el campo (Figura 5).

```
# El formato de los datos de la columna explicit, es del tipo booleano, sin embargo, algunos de los datos son de tipo caracter.
print(df['explicit'].value_counts())

# Debido esto genera que registros que se encuentran repetidos sean tomados como diferentes.
df[(df['album_name'] == 'Lover') &  (df['track_name'] == 'Cruel Summer')][['explicit', 'album_name', 'track_name']].sort_values(by= 'track_name')
```

✓ 0.0s

```
False    480
True      54
No         4
Si         1
Name: explicit, dtype: int64
```

Figura 5. Registros de tipo cadena de caracteres en columna del tipo booleano.

Las características del tipo flotante (float) *audio_features.danceability, audio_features.energy, audio_features.key, audio_features.loudness, audio_features.speechiness, y audio_features.liveness*, presentan registros catalogados como NaN (valores no numéricos) ya que se encuentran vacíos en el set de datos (Figura 6).

```
# Caracteristicas del tipo Float con registros NaN (valores no numéricos).
for feature in df[['audio_features.energy','audio_features.key','audio_features.loudness','audio_features.speechiness','audio_features.acousticness','audio_features.instrumentalness',
                   'audio_features.liveness','audio_features.valence','audio_features.tempo','audio_features.time_signature']]:
    display(df[feature][df[feature].isnull()])
```

✓ 0.0s

```
330   NaN
363   NaN
Name: audio_features.energy, dtype: float64

334   NaN
Name: audio_features.key, dtype: float64

330   NaN
334   NaN
Name: audio_features.loudness, dtype: float64

330   NaN
Name: audio_features.speechiness, dtype: float64

Series([], Name: audio_features.acousticness, dtype: object)

Series([], Name: audio_features.instrumentalness, dtype: object)

341   NaN
Name: audio_features.liveness, dtype: float64

Series([], Name: audio_features.valence, dtype: float64)

Series([], Name: audio_features.tempo, dtype: object)

363   NaN
Name: audio_features.time_signature, dtype: float64
```

Figura 6. Registros vacíos en columna del tipo flotante.

Por otra parte, la característica *audio_features.time_signature* que es del tipo entero (int) presenta un valor no numérico (index: 363) definido del tipo flotante (Figura 6).

Además, características del tipo flotante como danceability, acousticness, instrumentallness y tempo incluyen registros con tipo de formatos diferentes al definido para el campo (Figura 7).

```
# Tipo de dato erroneo en caracteristicas del tipo flotante
for feature in df[['audio_features.danceability', 'audio_features.acousticness', 'audio_features.instrumentalness', 'audio_features.tempo']].columns:

    print('')
    print(f'columna {feature}:')
    print('')

    for i, j  in enumerate(df[feature]):

        if feature == 'audio_features.instrumentalness' and j == 0:
            pass
        elif type(j) != float:
            print(f'index: {i} type: {type(j)} formato erroneo')
```

```
columna audio_features.danceability:

index: 330 type: <class 'NoneType'> formato erroneo
index: 431 type: <class 'str'> formato erroneo

columna audio_features.acousticness:

index: 1 type: <class 'int'> formato erroneo
index: 431 type: <class 'str'> formato erroneo
index: 535 type: <class 'int'> formato erroneo

columna audio_features.instrumentalness:

index: 524 type: <class 'str'> formato erroneo

columna audio_features.tempo:

index: 432 type: <class 'str'> formato erroneo
```

Figura 7. Registros de diferente tipo para columnas designadas como flotantes.

Es necesario tener en cuenta que la característica *audio_features.instrumentallness* presenta 235 registros con valor 0, los cuales son definidos como valores enteros. Aunque dichos valores parecieran ser errores de los datos, de acuerdo con la documentación de Spotify, para la columna *audio_features.instrumentallness* no hay un rango definido, por ende, no son catalogados como registros incorrectos.

De igual forma La columna *album_total_tracks* del tipo entero presenta valores de cadena de caracteres (str) (Figura 8).

```
cantidad = 0
for i, j in enumerate(df['album_total_tracks']):
    if type(j) != int:
        cantidad +=1
        print(f'index: {i} type: {type(j)} formato erroneo')
print(f'Cantidad de registros con el formato erroneo: {cantidad}')
```

✓ 0.0s

```
index: 524 type: <class 'str'> formato erroneo
index: 525 type: <class 'str'> formato erroneo
index: 526 type: <class 'str'> formato erroneo
index: 527 type: <class 'str'> formato erroneo
index: 528 type: <class 'str'> formato erroneo
index: 529 type: <class 'str'> formato erroneo
index: 530 type: <class 'str'> formato erroneo
index: 531 type: <class 'str'> formato erroneo
index: 532 type: <class 'str'> formato erroneo
index: 533 type: <class 'str'> formato erroneo
index: 534 type: <class 'str'> formato erroneo
index: 535 type: <class 'str'> formato erroneo
index: 536 type: <class 'str'> formato erroneo
index: 537 type: <class 'str'> formato erroneo
index: 538 type: <class 'str'> formato erroneo
Cantidad de registros con el formato erroneo: 15
```

Figura 8. Anomalías del tipo cadena de caracteres en columna del tipo entero.

Los errores relacionados con los registros de las columnas de formato cadena de caracteres están relacionados a registros faltantes del tipo NoneType (Figura 9), descritos previamente en la dimensión de completitud (Figura 2).

A su vez, la característica *track_id* tiene 4 registros del tipo cadena de caracteres, pero vacíos. Estos datos son normalmente de 22 caracteres (Figura 10).

Finalmente, características con rango definido como *audio_features.acousticness* y *artist_popularity* presentan 5 y 539 registros que exceden los límites permitidos, respectivamente (Figura 11).

```python
# Tipo de dato erroneo en caracteristicas del tipo string
for feature in df[['audio_features.id', 'track_id', 'track_name', 'audio_features.id', 'artist_id', 'artist_name', 'album_id', 'album_name', 'album_release_date']].columns:
    print('')
    print(f'columna {feature}:')

    for i, j  in enumerate(df[feature]):
        if type(j) != str:
            print(f'index: {i} type: {type(j)} formato erroneo')
```

✓ 0.0s

```
columna audio_features.id:

columna track_id:
index: 363 type: <class 'NoneType'> formato erroneo
index: 375 type: <class 'NoneType'> formato erroneo
index: 379 type: <class 'NoneType'> formato erroneo
index: 382 type: <class 'NoneType'> formato erroneo

columna track_name:
index: 77 type: <class 'NoneType'> formato erroneo
index: 91 type: <class 'NoneType'> formato erroneo
index: 104 type: <class 'NoneType'> formato erroneo

columna audio_features.id:
```

Figura 9. Campos vacíos en columnas del tipo cadena de caracteres.

```python
# Los registros de track_id son de 22 caracteres, algunos registros son del tipo string pero se encuentran vacíos.
for no, record in enumerate(df['track_id']):
    try:
        if len(record) < 22:
            print(f'Index: {no}, registro: {record}')
    except:
        pass
```

✓ 0.0s

```
Index: 321, registro:
Index: 434, registro:
Index: 442, registro:
Index: 445, registro:
```

Figura 10. Campos del formato cadena de caracteres pero vacíos.

```python
# Registros por fuera del rango apropiado.
for feature in df[['audio_features.acousticness', 'artist_popularity', 'audio_features.danceability', 'audio_features.key', 'audio_features.energy',
                   'audio_features.valence', 'audio_features.time_signature']]:
    print('')
    print(f'columna {feature}:')
    for no, record in enumerate(df[feature]):
        try:
            if feature in ['audio_features.danceability',  'audio_features.energy', 'audio_features.acousticness', 'audio_features.valence']:
                if record < 0 or record > 1:
                    print(f'index: {no} valor: {record} rango erroneo')
            elif feature == 'audio_features.key':
                if record < -1 or record > 11:
                    print(f'index: {no} valor: {record} rango erroneo')
            elif feature == 'audio_features.time_signature':
                if record < 3 or record > 7:
                    print(f'index: {no} valor: {record} rango erroneo')
            else:
                if record < 0 or record > 100:
                    print(f'index: {no} valor: {record} rango erroneo')
        except:
            pass
```

✓ 0.0s

```
columna audio_features.acousticness:
index: 1 valor: 5 rango erroneo
index: 3 valor: -0.000537 rango erroneo
index: 6 valor: -0.00354 rango erroneo
index: 527 valor: 1.5 rango erroneo
index: 535 valor: 2 rango erroneo

columna artist_popularity:
index: 0 valor: 120 rango erroneo
```

Figura 11. Registros por fuera del rango definido para las diferentes características.

A partir de los errores anteriormente expuestos, la tabla con anomalías para la dimensión validez es de la manera siguiente (Tabla 4):

Tabla 4. Anomalías encontradas para la dimensión validez.

Característica	No. de anomalías
explicit	5
track_id	7
track_name	3
**danceability*	2
**acousticness*	5
**instrumentalness*	1
**tempo*	1
artist_popularity	539
album_name	46

Las características marcadas con un (*) presentan los siguientes caracteres “audio_features.” previo al nombre del campo.

3.4. Precisión

Para la dimensión de Precisión de los datos se evidenciaron los siguientes errores (Tabla 5):

Tabla 5. Anomalías encontradas para la dimensión precisión.

Característica	No. de anomalías
album_release_date	39
album_total_tracks	112

La cantidad real de canciones por álbum no coincide con el valor de la columna designada para dicho valor. Álbumes como Red (taylor’s versión), evermore , Lover, reputation, y Taylor Swift presentan una cantidad diferente de canciones con relación al valor reportado en el set de datos (Figura 12).

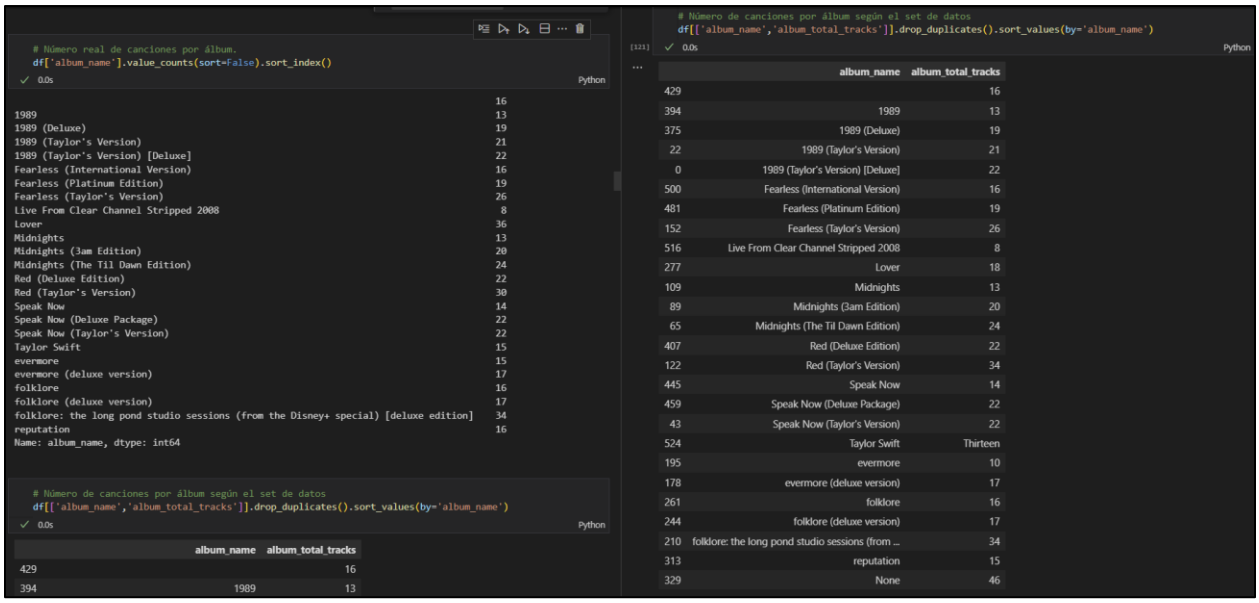

Figura 12**.** A) Cantidad de canciones reales por álbum. B) Cantidad de canciones por álbum reportadas en los datos.

Adicionalmente, de acuerdo con la información de la API de Spotify, los álbumes de Taylor Swift fueron estrenados desde el 2006 hasta el presente, por lo cual, la fecha reportada para el álbum “Midnights (The Til Dawn Edition)” es incorrecta ya que es superior a la actualidad. Así mismo, la fecha de estreno del

álbum Taylor Swift , 1984 según el registro, es incorrecta con relación a la fecha de lanzamiento real en 2006 (Figura 13).

```
print('Numero de registros con fechas de estreno del album por fuera de las fechas reales de lanzamiento:')
print(df[['album_release_date', 'album_name', 'album_id', 'track_name']][(df['album_release_date'].apply(lambda x: datetime.strptime(x, '%Y-%m-%d').year) < 2006) |
                                                                          (df['album_release_date'].apply(lambda x: datetime.strptime(x, '%Y-%m-%d').year) > 2024)].shape[0])
print('Algunos ejemplos son:')
df[['album_release_date', 'album_name', 'album_id', 'track_name']][(df['album_release_date'].apply(lambda x: datetime.strptime(x, '%Y-%m-%d').year) < 2006) |
                                                                   (df['album_release_date'].apply(lambda x: datetime.strptime(x, '%Y-%m-%d').year) > 2024)].sample(10)
```

✓ 0.0s

```
Numero de registros con fechas de estreno del album por fuera de las fechas reales de lanzamiento:
39
Algunos ejemplos son:
```

	album_release_date	album_name	album_id	track_name
82	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	Glitch
75	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	Karma
77	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	None
85	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	Hits Different
529	1989-10-24	Taylor Swift	5eyZZoQEFQWRHkV2xgAeBw	The Outside
535	1989-10-24	Taylor Swift	5eyZZoQEFQWRHkV2xgAeBw	I'm Only Me When I'm With You
65	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	Lavender Haze
81	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	High Infidelity
83	2027-05-26	Midnights (The Til Dawn Edition)	1fnJ7k0bIlNfL1kVdNVW1A	Would've, Could've, Should've
537	1989-10-24	Taylor Swift	5eyZZoQEFQWRHkV2xgAeBw	A Perfectly Good Heart

Figura 13. Registros de álbumes con fechas de lanzamiento erróneas.

3.5. Consistencia

Para la dimensión de consistencia se utilizaron los registros que no presentan variación entre ellos en cuanto al formato y contenido, es decir, los registros que no presentan errores en cuanto a validez, unicidad y precisión (Tabla 6):

Tabla 6. Anomalías encontradas para la dimensión consistencia.

Característica	No. de anomalías	Característica	No. de anomalías
disc_number	19	**instrumentalness*	20
duration_ms	19	**liveness*	20
explicit	23	**valence*	19
track_number	19	**tempo*	20
track_popularity	19	**id*	19
track_id	26	**time_signature*	20
track_name	22	*artist_id*	19
**danceability*	21	*artist_name*	19
**energy*	21	*artist_popularity*	19
**key*	20	*album_id*	19
**loudness*	21	*album_name*	65
**mode*	20	*album_release_date*	57
**speechiness*	20	*álbum_total_tracks*	114
**acousticness*	24	-	-

Las características marcadas con un (*) presentan los siguientes caracteres “audio_features.” previo al nombre del campo.

6. Evaluación:

Es importante mencionar que el porcentaje de datos erróneos es calculado sobre el total de los registros (539) y no sobre el set de datos corregido al eliminar las filas repetidas (520) para evitar alteraciones en la información original suministrada.

Tabla 7. Evaluación de la calidad de los datos en porcentajes.

	Completitud	**Unicidad**	**Validez**	**Precisión**	**Consistencia**	**Total**	**Calidad características**
disc_number	100,0	96,5	100,0	100,0	96,5	98,6	Buena
duration_ms	100,0	96,5	100,0	100,0	96,5	98,6	Buena
explicit	100,0	96,5	99,1	100,0	95,7	98,3	Buena
track_number	100,0	96,5	100,0	100,0	96,5	98,6	Buena
track_popularity	100,0	96,5	100,0	100,0	96,5	98,6	Buena
track_id	98,7	96,5	99,3	100,0	95,2	98,1	Buena
track_name	99,4	96,5	99,4	100,0	95,9	98,3	Buena
**danceability*	99,8	96,5	99,6	100,0	96,1	98,4	Buena
**energy*	99,6	96,5	100,0	100,0	96,1	98,4	Buena
**key*	99,8	96,5	100,0	100,0	96,3	98,5	Buena
**loudness*	99,6	96,5	100,0	100,0	96,1	98,4	Buena
**mode*	100,0	96,5	100,0	100,0	96,3	98,6	Buena
**speechiness*	99,8	96,5	100,0	100,0	96,3	98,5	Buena
**acousticness*	100,0	96,5	99,1	100,0	95,5	98,2	Buena
**instrumentalness*	100,0	96,5	99,8	100,0	96,3	98,5	Buena
**liveness*	99,8	96,5	100,0	100,0	96,3	98,5	Buena
**valence*	100,0	96,5	100,0	100,0	96,5	98,6	Buena
**tempo*	100,0	96,5	99,8	100,0	96,3	98,5	Buena
**id*	100,0	0,0	100,0	100,0	96,5	79,3	Mala
**time_signature*	99,8	96,5	100,0	100,0	96,3	98,5	Buena
artist_id	100,0	96,5	100,0	100,0	96,5	98,6	Buena
artist_name	100,0	96,5	100,0	100,0	96,5	98,6	Buena
artist_popularity	100,0	96,5	0,0	100,0	96,5	78,6	Mala
album_id	100,0	96,5	100,0	100,0	96,5	98,6	Buena
album_name	91,5	96,5	99,3	100,0	87,9	95,0	Aceptable
album_release_date	100,0	96,5	100,0	92,8	89,4	95,7	Aceptable
álbum_total_tracks	100,0	96,5	100,0	79,2	78,8	90,9	Aceptable
Total	99,6	92,9	96,1	99,0	95,0	**96,5**	**Buena**
Calidad dimensión	Buena	Aceptable	Buena	Buena	Aceptable		

Las características marcadas con un (*) presentan los siguientes caracteres “audio_features.” Previo al nombre del campo.

7. Análisis de calidad de los datos:

En términos generales la calidad de los datos es buena, con un total de 96.5 sobre 100. Las diferentes dimensiones analizadas son catalogadas como buenas y aceptables (60 % y 40 % del total, respectivamente), siendo unicidad la dimensión con la calificación más baja (92.9 %).

Desde el contexto de las características, los datos de 22 de estas son caracterizadas como buenos, correspondientes a un 82.8 % de la información, 3 son definidos como aceptables (10.3 %) y las características *id* y *artist_popularity* clasificadas como malas (6.9 %).

8. Referencias:

DAMA UK. 2018. The six primary dimensions for data quality assessment. defining data quality dimensions.