Analisis Calidad de la data

```
import pyspark
from pyspark.sql.functions import *
```

Creamos la sesion spark y la inicializamos

```
spark = pyspark.sql.SparkSession.builder.appName("quality").getOrCreate()
```

Carga de base de datos

```
path = "dataset.csv"
df = spark.read.csv(path,header=True,sep=',', inferSchema="True")
df.show(5)
```

```
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+--------------------------+--
|disc_number|duration_ms|explicit|track_number|track_popularity|            track_id|          track_name|audio_features.danceability|au
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+--------------------------+--
|          1|     212600|   False|           1|              77|4WUepByoeqcedHoYh...|Welcome To New Yo...|                     0.757|
|          1|     231833|   False|           2|              78|0108kcWLnn2HlH2ke...|Blank Space (Tayl...|                     0.733|
|          1|     231000|   False|           3|              79|3Vpk1hfMAQme8VJ0S...|Style (Taylor's V...|                     0.511|
|          1|     235800|   False|           4|              78|1OcSfkeCg9hRC2sFK...|Out Of The Woods ...|                     0.545|
|          1|     193289|   False|           5|              77|2k0ZEeAqzvYMcx9Qt...|All You Had To Do...|                     0.588|
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+--------------------------+--
only showing top 5 rows
```

Lo primero es entender la base que estamos procesando

Tenemos una base de albunes con sus canciones de Taylor Swift asi mismo validamos las columnas donde se encuentran variables como que tan bailable puede ser, su energia , presencia de ruido , o palabras entre otros

```
df.printSchema()
```

```
root
 |-- disc_number: integer (nullable = true)
 |-- duration_ms: integer (nullable = true)
 |-- explicit: string (nullable = true)
 |-- track_number: integer (nullable = true)
 |-- track_popularity: integer (nullable = true)
 |-- track_id: string (nullable = true)
 |-- track_name: string (nullable = true)
 |-- audio_features.danceability: double (nullable = true)
 |-- audio_features.energy: double (nullable = true)
 |-- audio_features.key: double (nullable = true)
 |-- audio_features.loudness: double (nullable = true)
 |-- audio_features.mode: integer (nullable = true)
 |-- audio_features.speechiness: double (nullable = true)
 |-- audio_features.acousticness: double (nullable = true)
 |-- audio_features.instrumentalness: string (nullable = true)
 |-- audio_features.liveness: double (nullable = true)
 |-- audio_features.valence: double (nullable = true)
 |-- audio_features.tempo: double (nullable = true)
 |-- audio_features.id: string (nullable = true)
 |-- audio_features.time_signature: double (nullable = true)
 |-- artist_id: string (nullable = true)
 |-- artist_name: string (nullable = true)
 |-- artist_popularity: integer (nullable = true)
 |-- album_id: string (nullable = true)
 |-- album_name: string (nullable = true)
 |-- album_release_date: date (nullable = true)
 |-- album_total_tracks: string (nullable = true)
```

Comprobamos los tipos de cada columna que esten acorde a su significado dado que esto puede afectar para un modelado cuando una columna no tenga el topo correcto

en este caso vemos las columnas album_total_tracks y audio_features.instrumentalness: estan en un formato string , cuando deberia tener un formato integer y double respectivamente

para el caso de audio_features.mode este debneria ser un boleano dado que toma solo valores de "0" y 1

Valores nulos

Como tenemos nombres de columnas con ".", reemplazamos esto por "_" para poder hacer el conteo de los nulos por columnas y no tener conflictos en pyspark

```
df1 = df.toDF(*[c.replace('.', '_') for c in df.columns])
df1.show(3)
```

```
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+---------------------------+--
|disc_number|duration_ms|explicit|track_number|track_popularity|            track_id|          track_name|audio_features_danceability|au
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+---------------------------+--
|          1|     212600|   False|           1|              77|4WUepByoeqcedHoYh...|Welcome To New Yo...|                      0.757|
|          1|     231833|   False|           2|              78|0108kcWLnn2HlH2ke...|Blank Space (Tayl...|                      0.733|
|          1|     231000|   False|           3|              79|3Vpk1hfMAQme8VJ0S...|Style (Taylor's V...|                      0.511|
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+---------------------------+--
only showing top 3 rows
```

```
df1.select([count(when(col(k).isNull(),k)).alias(k) for k in df1.columns ]).show()
```

```
+-----------+-----------+--------+------------+----------------+--------+----------+---------------------------+---------------------+--
|disc_number|duration_ms|explicit|track_number|track_popularity|track_id|track_name|audio_features_danceability|audio_features_energy|au
+-----------+-----------+--------+------------+----------------+--------+----------+---------------------------+---------------------+--
|          0|          0|       0|           0|               0|       8|         7|                          2|                    2|
+-----------+-----------+--------+------------+----------------+--------+----------+---------------------------+---------------------+--
```

Observamos que donde tenemos mas datos faltanes es en el nombre del album

Resumen general

```
df1.describe().show()
```

```
+-------+-------------------+------------------+--------+------------------+------------------+--------------------+--------------------
|summary|        disc_number|       duration_ms|explicit|      track_number|  track_popularity|            track_id|          track_name
+-------+-------------------+------------------+--------+------------------+------------------+--------------------+--------------------
|  count|                539|               539|     539|               539|               539|                 531|                 532
|   mean| 1.0315398886827458| 236003.7254174397|    null|11.280148423005565| 62.91836734693877|                null|                22.0
| stddev|0.17493398591537432|55019.871010413415|    null| 7.965620550754272|22.498757014954524|                null|                 0.0
|    min|                  1|           -223093|   False|                 1|               -92|00vJzaoxM3Eja1doB...|"""Slut!"" (Taylo...
|    max|                  2|            613026|    True|                46|               152|7zcnlq38eqNWyUF6e...|‘tis the damn season
+-------+-------------------+------------------+--------+------------------+------------------+--------------------+--------------------
```

Podemos ver inconsistencia en duration_ms en donde tenemos valores minimos negativos, asi como en la variable audio_features_acousticness

Valores duplicados

```
duplicates = df1.groupBy(*df1.columns).count().filter(col("count") > 1)
duplicates.show()
duplicates.count()
```

```
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+--------------------------+--
|disc_number|duration_ms|explicit|track_number|track_popularity|            track_id|          track_name|audio_features_danceability|au
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+--------------------------+--
|          1|     173386|   False|           6|              78|2YWtcWi3a83pdEg3G...|    I Think He Knows|                     0.897|
|          1|     193000|   False|          16|              80|2Rk4JlNc2TPmZe2af...|ME! (feat. Brendo...|                      0.61|
|          1|     171360|   False|          14|              84|6RRNNciQGZEXnqk8S...|You Need To Calm ...|                     0.771|
|          1|     211240|   False|           5|              82|3pHkh7d0lzM2AldUt...|          The Archer|                     0.292|
|          1|     198533|   False|          10|              79|2dgFqt3w9xIQRjhPt...|Death By A Thousa...|                     0.712|
|          1|     223293|   False|          15|              82|1SymEzIT3H8UZfibC...|           Afterglow|                     0.756|
|          1|     234466|    True|          21|              82|3xYJScVfxByb61dYH...|      Hits Different|                     0.672|
|          1|     221306|   False|           3|              92|1dGr1c8CrMLDpV6mP...|               Lover|                     0.359|
|          1|     293453|   False|          18|              85|1fzAuUVbzlhZ1lJAx...|            Daylight|                     0.557|
|          1|     190360|   False|           4|              86|3RauEVgRgj1IuWdJ9...|             The Man|                     0.777|
|          1|     287266|   False|           9|              81|12M5uqx0ZuwkpLp5r...|     Cornelia Street|                     0.824|
|          1|     170640|   False|           1|              77|43rA71bccXFGD4C8G...|I Forgot That You...|                     0.664|
|          1|     200306|   False|          13|              78|5hQSXkFgbxjZo9uCw...|           False God|                     0.739|
|          1|     150440|   False|          17|              72|1SmiQ65iSAbPto6gP...|It’s Nice To Have...|                     0.737|
|          1|     201586|   False|          12|              72|4AYtqFyFbX0Xkc2wt...|Soon You’ll Get B...|                     0.433|
|          1|     234146|   False|           7|              83|214nt20w5wOxJnY46...|Miss Americana & ...|                     0.662|
|          1|     190240|   False|          11|              80|1LLXZFeAHK9R4xUra...|          London Boy|                     0.695|
|          1|     222400|   False|           8|              86|4y5bvROuBDPr5fuwX...|         Paper Rings|                     0.811|
+-----------+-----------+--------+------------+----------------+--------------------+--------------------+--------------------------+--

18
```

Aca podemos observar que tenemos 18 filas duplicadas , los cuales contienen los mismo registros para todas las columnas

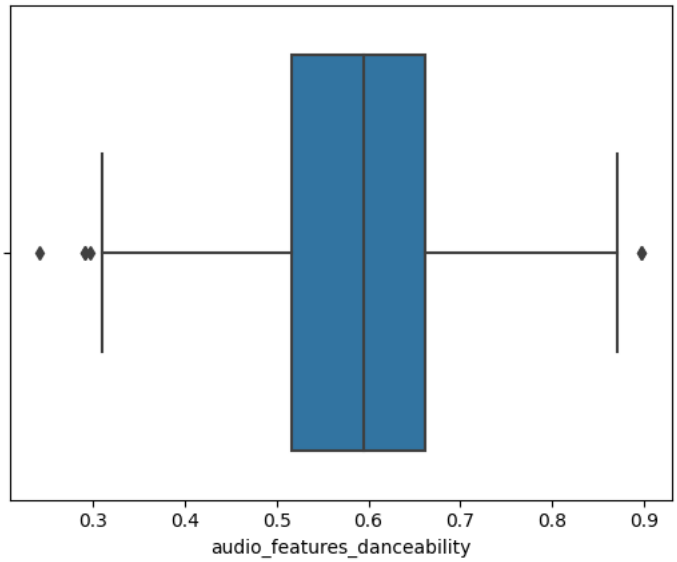
Alternativa grafica

```
import pandas as pd
import matplotlib.pyplot  as plt
import seaborn as sns
```

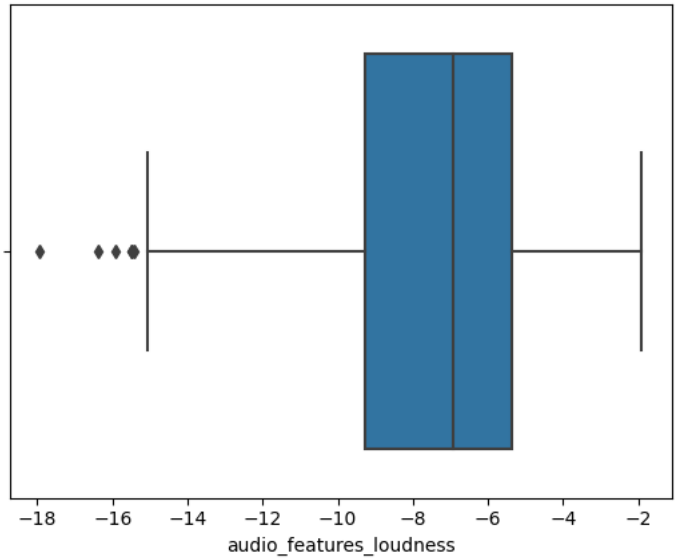
```
pandas_df = df1.toPandas()
```

Mediante un boxplot podemos ver tambien que valores atipicos tienen las variables , con respecto a la media y la sd, esto nos da una mayor visual parakl poder ir revisando la calidad de la data

```
sns.boxplot(x='audio_features_danceability', data=pandas_df)
plt.show()
```



```
sns.boxplot(x='audio_features_loudness', data=pandas_df)
plt.show()
```




```
sns.boxplot(x='audio_features_tempo', data=pandas_df)
plt.show()
```