Ficha técnica: proyecto 2 Hipótesis

Estado	Completado

Ficha técnica: proyecto 2 : Hipótesis

Titulo del Proyecto: Proyecto Musical Spotify

- **Objetivo**

Mediante la preparación y limpieza de la base de datos proporcionar recomendaciones estratégicas con técnicas de segmentación como el RFM y la validación de Hipótesis para que la discográfica y el nuevo artista puedan tomar decisiones informadas que aumenten sus posibilidades de conseguir el “éxito”.

- **Equipo**

Karina Diaz

Ingrid Carolina Ruiz

- **Herramientas y Tecnologías**

BigQuery (SQL)

Power BI (Visualización de tablas y graficas)

Python

Google Slides

Loom

Git Hub

- **Procesamiento y análisis**

Hito 1

Procesar y preparar la base de datos :

Conectar/importar datos a otras herramientas:

Se importo a BigQuery 3 tablas: track_in_comprtition, track_in_spotify y track_technical_info.

Identificar y manejar valores nulos:

Se identificaron 50 nulos en la variable shazam chart de la tabla track_in_competition

```
SELECT
  COUNT (*)
FROM
  `spotify-428200.dataset.track_in_competition`
WHERE
  track_id IS NULL
  OR in_apple_playlists IS NULL
  OR in_apple_charts IS NULL
  OR in_deezer_playlists IS NULL
  OR in_deezer_charts IS NULL
  OR in_shazam_charts IS NULL
```

Se identifico 95 en la variable Key de la tabla track_technical_info

```
SELECT
  COUNT (*)
FROM
  `spotify-428200.dataset.track_technical_info`
WHERE
  track_id IS NULL
  OR bpm IS NULL
  OR KEY IS NULL
  OR mode IS NULL
  OR 'danceability_%' IS NULL
  OR 'valance_%' IS NULL
  OR 'energy_%' IS NULL
  OR 'acoustiness_%' IS NULL
  OR 'instrumentalness_%' IS NULL
  OR 'liveness_%' IS NULL
  OR 'speechiness_%' IS NULL
```

Identificar y manejar valores duplicados:

se identifico duplicados en la tabla track_in_spotify y se utilizo la siguiente consulta:

```
SELECT
  track_name,
  artist_s__name,
  COUNT(*) AS cantidad
FROM
  `spotify-428200.dataset.track_in_spotify`
GROUP BY
  track_name,
  artist_s__name
HAVING
  COUNT(*) > 1;
```

Identificar y manejar datos fuera del alcance del análisis:

se identifico valores como Key y modo que no nos eran relevantes en este proyecto en la tabla track_technical_info y se uso la siguiente consulta:

```
SELECT
  * EXCEPT (key, mode)
FROM
  `spotify-428200.dataset.track_technical_info`
```

Identificar y manejar datos discrepantes en variables categóricas:

Se identifico estos valores en la tabla track_in_Spotify y se uso la siguiente consulta:

```
SELECT
  track_id,
  REGEXP_REPLACE(track_name, r'[^a-zA-z0-9 ]', ' ') AS track_
  REGEXP_REPLACE(artist_s__name,r'[^a-zA-z0-9 ]', ' ') AS art
  artist_count,
  released_year,
  released_month,
  released_day,
  in_spotify_playlists,
  in_spotify_charts
```

```
FROM
  `spotify-428200.dataset.track_in_spotify`
```

Identificar y manejar datos discrepantes en variables numéricas:

Se identifico estas variables en la tabla track_in_spotify y luego se regresa para hacerlo correctamente en la nueva tabla modificada con la siguiente consulta:

```
SELECT
MAX(streams),
MIN(streams),
AVG(streams)
FROM `spotify-428200.dataset.track_in_spotify_modificado`
```

Comprobar y cambiar tipo de dato:

Se cambio dato streams de string a integer y se tomo la decisión de excluir algunos tracks en la tabla track_in_spotify_modificado con la siguiente consulta:

```
CREATE TABLE
  `spotify-428200.dataset.track_in_spotify_modificado` AS
SELECT
  track_id,
  track_name,
  artist_s__name,
  artist_count,
  released_year,
  released_month,
  released_day,
  in_spotify_playlists,
  in_spotify_charts,
  SAFE_CAST(streams AS INT64) AS streams
FROM
  `spotify-428200.dataset.track_in_spotify`
WHERE
  SAFE_CAST(streams AS INT64) IS NOT NULL
AND track_id NOT IN ('5080031', '8173823', '1119309', '381467
```

Crear nuevas variables

Se creo una nueva variable released_date con la siguiente consulta

```
SELECT
  CONCAT(
    CAST(released_year AS STRING), '-',
    LPAD(CAST(released_month AS STRING), 2, '0'), '-',
    LPAD(CAST(released_day AS STRING), 2, '0')
  ) AS release_date
FROM
  `spotify-428200.dataset.track_in_spotify`
```

Unir tablas

Se crearon tablas views antes de hacer la union completa con Left Join

```
SELECT
  track_id,
  in_apple_playlists,
  in_apple_charts,
  in_deezer_playlists,
  in_deezer_charts,
  in_shazam_charts,
  (IFNULL(in_apple_playlists, 0) + IFNULL(in_deezer_playlists
  (IFNULL(in_apple_charts, 0) + IFNULL(in_deezer_charts, 0) +
FROM
  `spotify-428200.dataset.track_in_competition_modificado`
WHERE track_id NOT IN ('5080031', '8173823', '1119309', '3814
```

```
SELECT
  track_id,
  REGEXP_REPLACE(track_name, r'[^a-zA-Z0-9 ]', '') AS track_n
  REGEXP_REPLACE(artist_s__name,r'[^a-zA-z0-9 ]', ' ') AS art
  artist_count,
  released_year,
  released_month,
  released_day,
  in_spotify_playlists,
  in_spotify_charts,
```

```
  SAFE_CAST(streams AS INT64) AS streams,
  CONCAT(
    CAST(released_year AS STRING), '-',
    LPAD(CAST(released_month AS STRING), 2, '0'), '-',
    LPAD(CAST(released_day AS STRING), 2, '0')
  ) AS release_date
FROM
  `spotify-428200.dataset.track_in_spotify_modificado`
WHERE
  SAFE_CAST(streams AS INT64) IS NOT NULL
  AND track_id NOT IN ('5080031', '8173823', '1119309', '3814
```

```
SELECT
 * EXCEPT (key, mode)
  FROM `spotify-428200.dataset.track_technical_info`
  WHERE
  track_id NOT IN ('5080031', '8173823', '1119309', '3814670'
```

```
SELECT *
FROM `spotify-428200.dataset.view_track_in_spotify`
LEFT JOIN `spotify-428200.dataset.view_track_in_competition`
USING (track_id)
LEFT JOIN `spotify-428200.dataset.view_track_technical_info`
USING (track_id)
```

Construir tablas auxiliares

Se construyo esta tabla con el comando WITH para ver el total en playlist de las canciones por artista con la siguiente consulta:

```
WITH total_canciones_por_artista AS (
  SELECT
    artist_s__name_limpio,
    COUNT(track_id) AS total_canciones
  FROM
    `spotify-428200.dataset.view_complete_table`
  GROUP BY
    artist_s__name_limpio
```

```
)
SELECT
  a.*,
  t.total_canciones
FROM
  `spotify-428200.dataset.view_complete_table` a
LEFT JOIN
  total_canciones_por_artista t
ON
  a.artist_s__name_limpio = t.artist_s__name_limpio;
```

- **Hacer un análisis exploratorio**

Agrupar datos según variables categóricas

Se realizo la importación de la tabla view_complete_table a Power Bi y se procedió a empezar a hacer una tabla Matriz con las variables Artista y total_track y released_year por total_track.

Visualizar las variables categóricas

Se crearon gráficos de barras apiladas para mostrar los resultados visibles de las variables anteriores

Aplicar medidas de tendencia central

Se realizo el calculo de promedio, mediana, mínimo y máximo de las variables bpm, streams, total_playlist y spotify playlist.

Visualizar distribución

Se crearon histogramas por medio de python para visualizar la distribución de las variables de streams y spotify_playlist

Aplicar medidas de dispersión

Se calcula la desviación estándar de las variables bpm, streams, total_playlist y in_spotify_playlist.

Visualizar el comportamiento de los datos a lo largo del tiempo

Se crearon gráficos en líneas de las variables track_name_limpio con released_date y streams con released_date

Calcular cuartiles, deciles o percentiles

Regresamos nuevamente a BigQuery y calculamos los quartiles de streams y los categóricos, a continuación comparto la siguiente consultas:

```
WITH
  Quartiles AS (
  SELECT
    streams,
    NTILE(4) OVER (ORDER BY streams) AS quartile_streams
  FROM
    `dataset.view_complete_table` )
SELECT
  a.*,
  Quartiles.quartile_streams
FROM
  `dataset.view_complete_table` a
LEFT JOIN
  Quartiles
ON
  a.streams=Quartiles.streams
```

```
WITH Quartiles AS (
  SELECT
    track_id,
    streams,
    bpm,
    `danceability_%`,
    `valence_%`,
    `energy_%`
    `acousticness_%`,
    `instrumentalness_%`,
    `liveness_%`,
    `speechiness_%`,
    NTILE(4) OVER (ORDER BY streams) AS quartile_streams,
    NTILE(4) OVER (ORDER BY bpm) AS quartile_bpm,
    NTILE(4) OVER (ORDER BY `danceability_%`) AS quartile_dan
    NTILE(4) OVER (ORDER BY `valence_%`) AS quartile_valence,
    NTILE(4) OVER (ORDER BY `energy_%`) AS quartile_energy,
    NTILE(4) OVER (ORDER BY `acousticness_%`) AS quartile_aco
```

```
    NTILE(4) OVER (ORDER BY `instrumentalness_%`) AS quartile
    NTILE(4) OVER (ORDER BY `liveness_%`) AS quartile_livenes
    NTILE(4) OVER (ORDER BY `speechiness_%`) AS quartile_spee
  FROM `dataset.view_complete_table`
)
SELECT
   a.*,
   CASE
       WHEN QUARTILES.quartile_streams = 1 THEN 'bajo'
       WHEN QUARTILES.quartile_streams = 2 THEN 'medio bajo'
       WHEN QUARTILES.quartile_streams = 3 THEN 'medio'
       WHEN QUARTILES.quartile_streams = 4 THEN 'alto'
   END AS cat_streams,
   CASE
       WHEN QUARTILES.quartile_bpm = 1 THEN 'lento'
       WHEN QUARTILES.quartile_bpm = 2 THEN 'moderado'
       WHEN QUARTILES.quartile_bpm = 3 THEN 'rápido'
       WHEN QUARTILES.quartile_bpm = 4 THEN 'muy rápido'
   END AS cat_bpm,
   CASE
       WHEN QUARTILES.quartile_danceability = 1 THEN 'bajo'
       WHEN QUARTILES.quartile_danceability = 2 THEN 'medio b
       WHEN QUARTILES.quartile_danceability = 3 THEN 'medio'
       WHEN QUARTILES.quartile_danceability = 4 THEN 'alto'
   END AS cat_danceability,
   CASE
       WHEN QUARTILES.quartile_valence = 1 THEN 'bajo'
       WHEN QUARTILES.quartile_valence = 2 THEN 'medio bajo'
       WHEN QUARTILES.quartile_valence = 3 THEN 'medio'
       WHEN QUARTILES.quartile_valence = 4 THEN 'alto'
   END AS cat_valence,
   CASE
       WHEN QUARTILES.quartile_energy = 1 THEN 'bajo'
       WHEN QUARTILES.quartile_energy = 2 THEN 'medio bajo'
       WHEN QUARTILES.quartile_energy = 3 THEN 'medio'
       WHEN QUARTILES.quartile_energy = 4 THEN 'alto'
   END AS cat_energy,
   CASE
```

```
        WHEN QUARTILES.quartile_acousticness = 1 THEN 'bajo'
        WHEN QUARTILES.quartile_acousticness = 2 THEN 'medio b
        WHEN QUARTILES.quartile_acousticness = 3 THEN 'medio'
        WHEN QUARTILES.quartile_acousticness = 4 THEN 'alto'
    END AS cat_acousticness,
    CASE
        WHEN QUARTILES.quartile_instrumentalness = 1 THEN 'baj
        WHEN QUARTILES.quartile_instrumentalness = 2 THEN 'med
        WHEN QUARTILES.quartile_instrumentalness = 3 THEN 'med
        WHEN QUARTILES.quartile_instrumentalness = 4 THEN 'alt
    END AS cat_instrumentalness,
    CASE
        WHEN QUARTILES.quartile_liveness = 1 THEN 'bajo'
        WHEN QUARTILES.quartile_liveness = 2 THEN 'medio bajo'
        WHEN QUARTILES.quartile_liveness = 3 THEN 'medio'
        WHEN QUARTILES.quartile_liveness = 4 THEN 'alto'
    END AS cat_liveness,
    CASE
        WHEN QUARTILES.quartile_speechiness = 1 THEN 'bajo'
        WHEN QUARTILES.quartile_speechiness = 2 THEN 'medio ba
        WHEN QUARTILES.quartile_speechiness = 3 THEN 'medio'
        WHEN QUARTILES.quartile_speechiness = 4 THEN 'alto'
    END AS cat_speechiness
FROM `dataset.view_complete_table` a
LEFT JOIN Quartiles
ON a.track_id = Quartiles.track_id;
```

Calcular correlación entre variables

Calculamos las siguientes variables en BiGQuery para observar la correlacion de Pearson entre ambas con esta consulta:

```
SELECT
  CORR(streams, bpm) AS correlation_value_bpm,
  CORR(streams, total_en_playlist) AS correlation_value_strea
  CORR(streams, in_spotify_playlists) AS correlation_value_st
  CORR(streams, `danceability_%`) AS correlation_value_stream
  CORR(streams,`valence_%`) AS correlation_value_streams_vale
  CORR(streams,`energy_%`) AS correlation_value_streams_energ
```

```
  CORR(streams,`acousticness_%`) AS correlation_value_streams
  CORR(streams,`instrumentalness_%`) AS correlation_value_str
  CORR(streams,`liveness_%`) AS correlation_value_streams_liv
  CORR(streams,`speechiness_%`) AS correlation_value_streams_
FROM
  `spotify-428200.dataset.view_complete_table`
```

- **Aplicar técnica de análisis**

Aplicar segmentación

Creación de una tabla matrix en Power BI para ver el promedio de streams por categorias (alto y bajo) de cada característica de la canción (danceability_%**, valence_%**, energy_%**, acousticness_%**, instrumentalness_%**, liveness_%**, speechiness_%)

Validar Hipótesis
Refutar las cinco hipotesis planteadas a traves del calculo de la correlación de pearson que se realizp en BigQuery y visualizarlas a través de un gráficos en Power BI para comprender y analizar lo que hace una canción exitosa.

- **Resultados y Conclusiones**

A continuación se presenta algunos datos del cual nos basamos para realizar nuestro análisis:

1- La discográfica se enfrenta al emocionante desafío de lanzar un nuevo artista en el escenario musical global.

2- Nos proporciona una herramienta poderosa un extenso dataset de Spotify con información sobre las canciones más escuchadas en 2023.

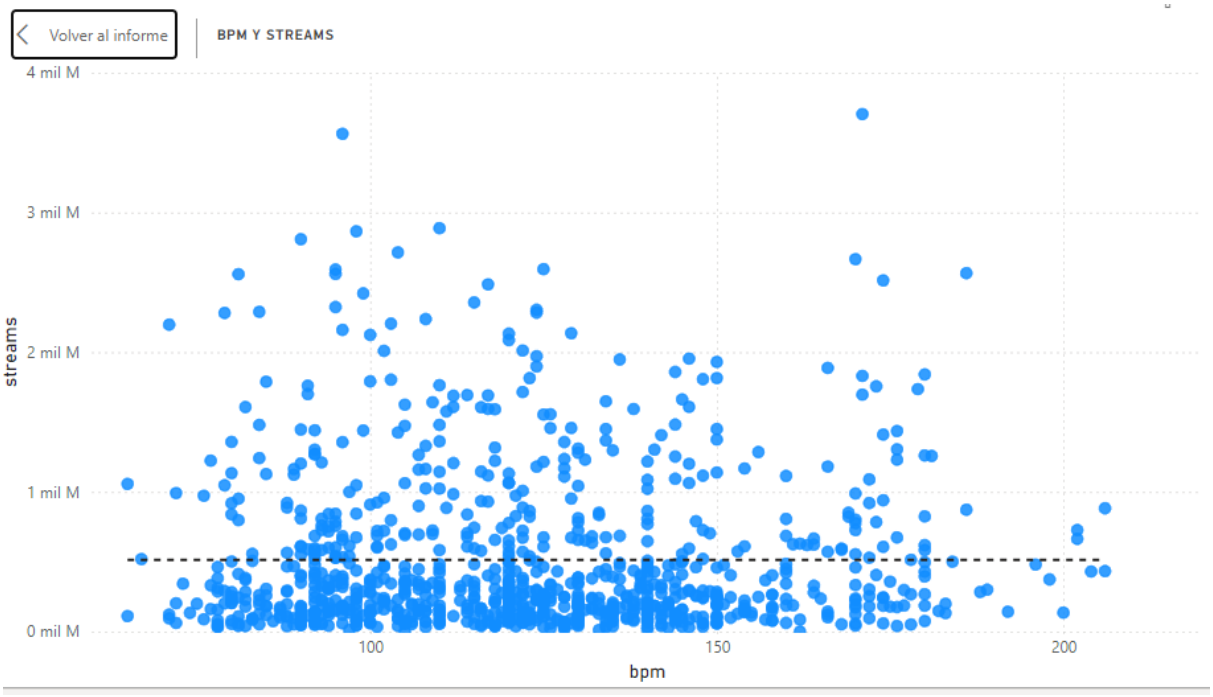
3-Tenemos que validar o refutar una serie de hipótesis mediante técnica de análisis para proporcionar al cliente estrategias para que ellos puedan tomar una mejor toma de decisión y aumente sus posibilidades de conseguir el éxito.

Hipótesis 1

Las canciones con un mayor BPM (Beats Por Minuto) tienen más éxito en términos de streams en Spotify

Correlación negativa:

- -0.0023050669108 significa que no existe una relación entre ambas variables de los beats o pulsaciones por minutos y la cantidad reproducción en una canción en playlist.

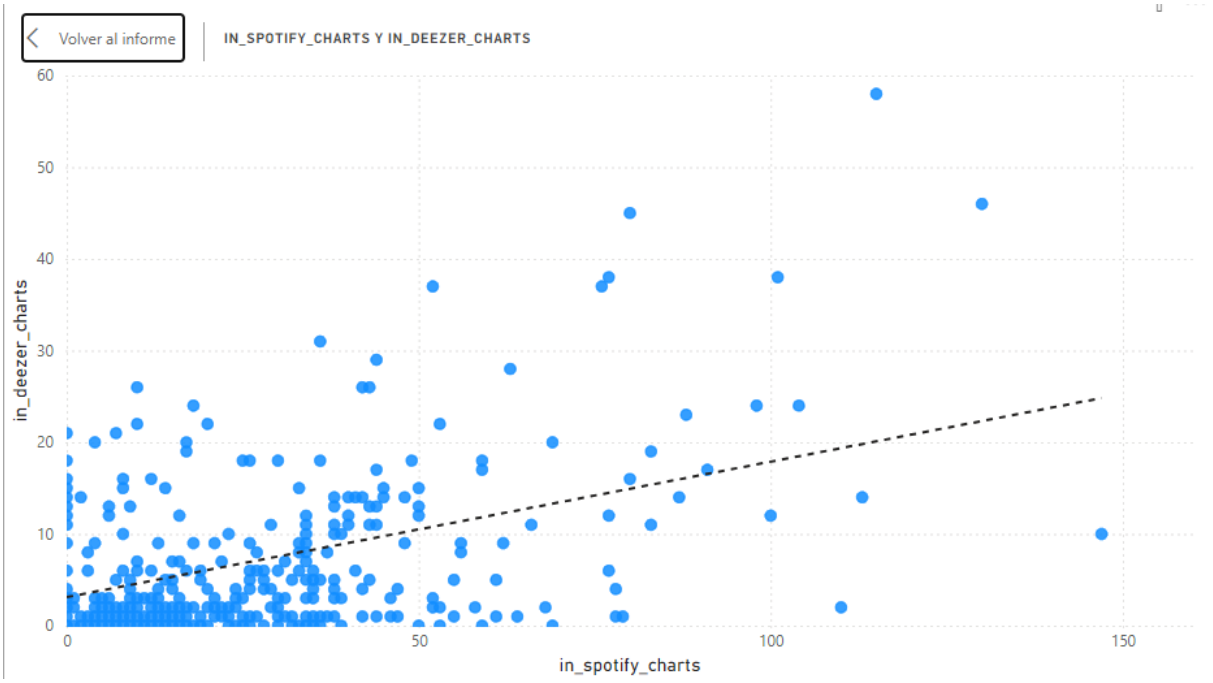


Hipótesis 2

Las canciones más populares en el ranking de Spotify también tienen un comportamiento similar en otras plataformas como Deezer.

Correlación positiva:

0.5996905941510716 2 lo que significa que ambas variables de in_spotify_charts e in_deezer_charts tienen popularidad de una canción entre el público en estas plataformas digitales.

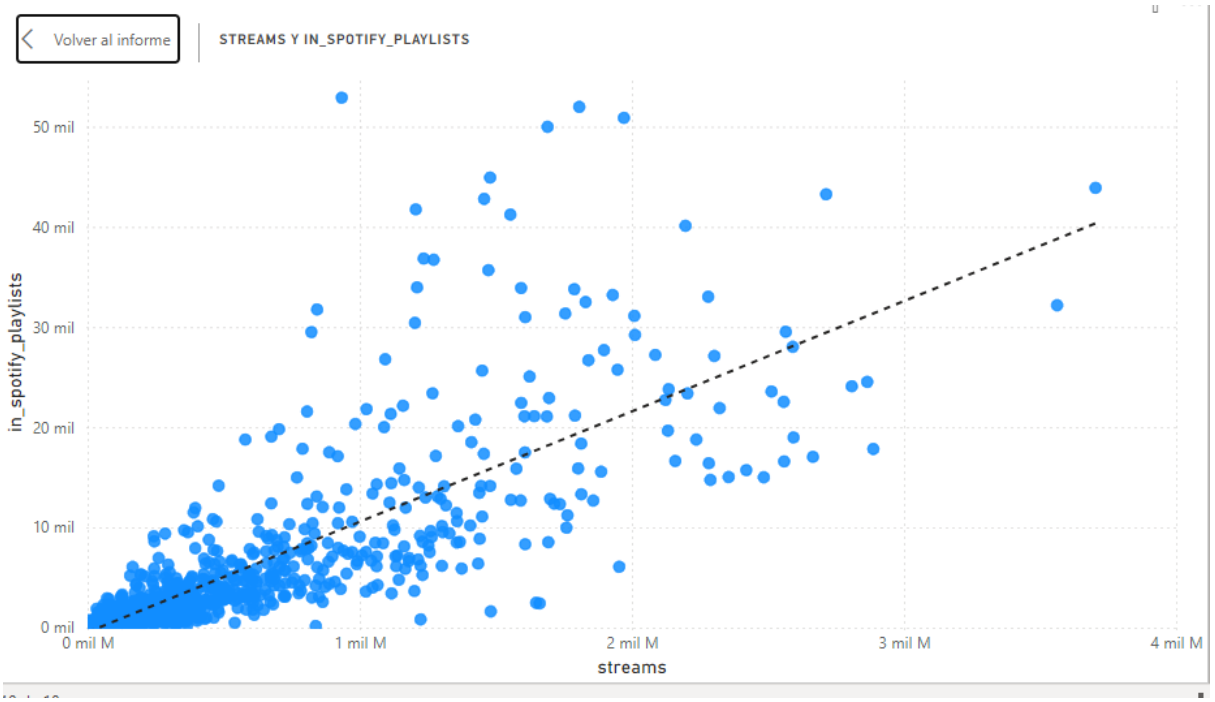


Hipótesis 3

La presencia de una canción en un mayor número de playlists se relaciona con un mayor número de streams.

Correlación positiva: 0.790330199938372

lo que indica que indica que si la canción está en un mayor número de playlists más aumenta los streams como es el caso de la plataforma spotify.

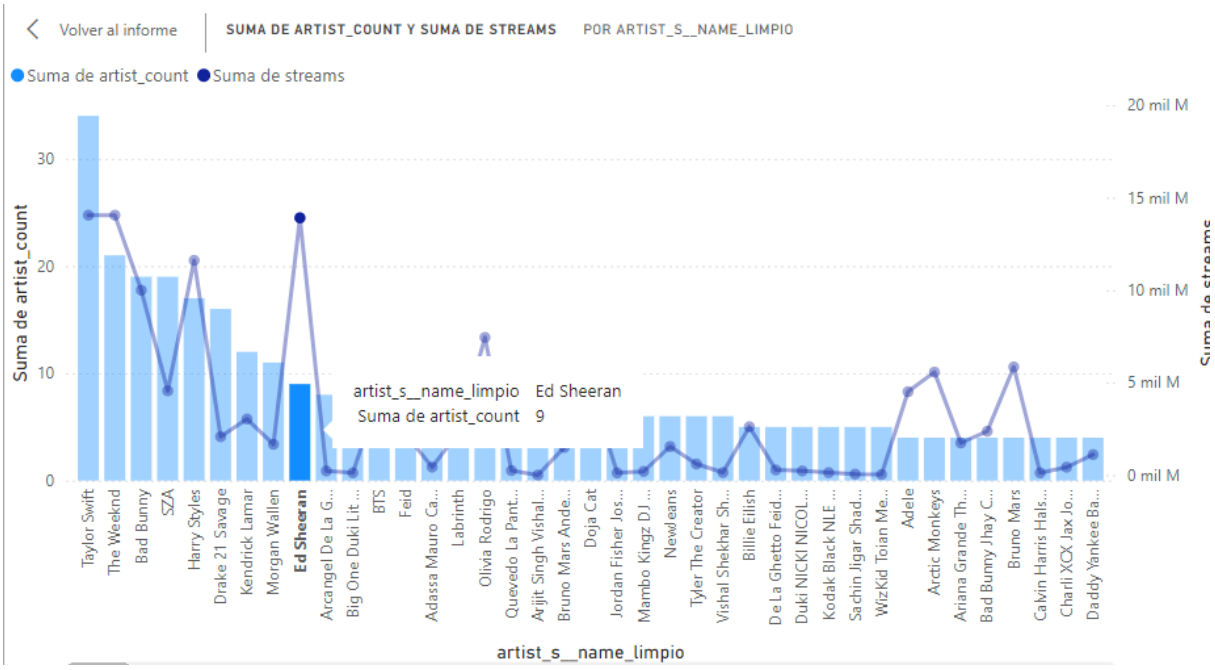


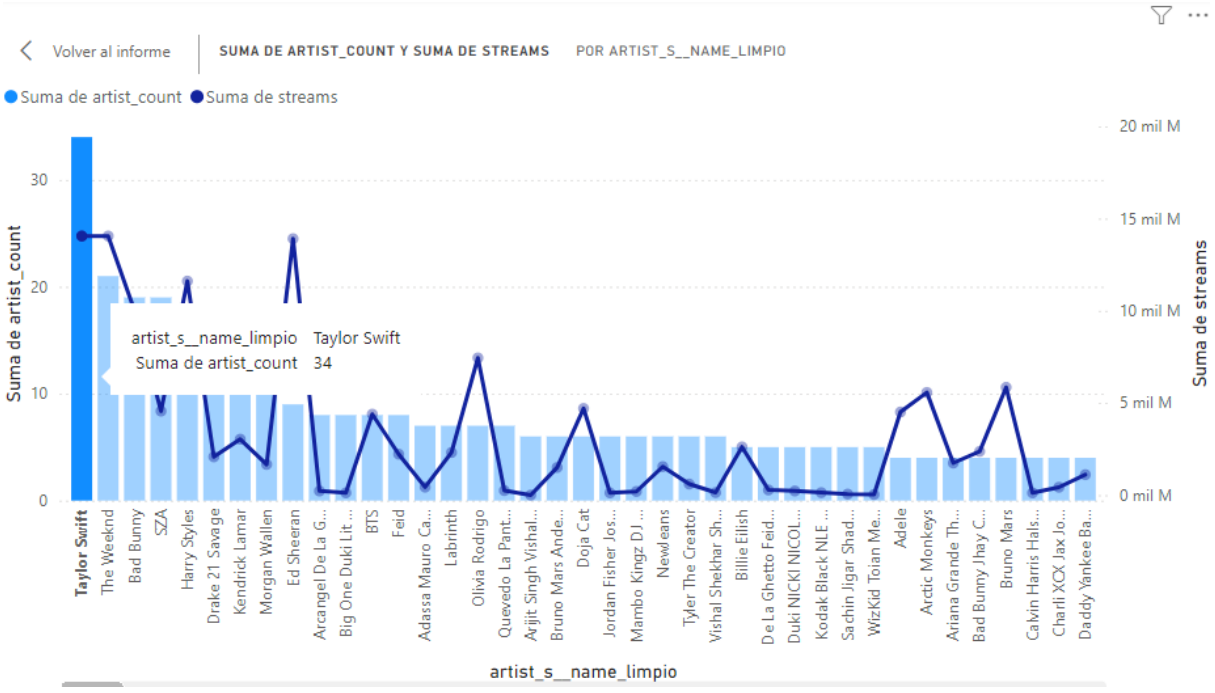
Hipótesis 4

Los artistas con un mayor número de canciones en Spotify tienen más streams.

Correlación negativa: -0.1368564661718

En este caso no siempre los artista con mayor número de canciones tienen mas streams, en la esta grafica podemos ver que Ed Sheeran con solo 9 canciones tiene casi la misma cantidad de streams que Taylor Swift con 34 canciones.





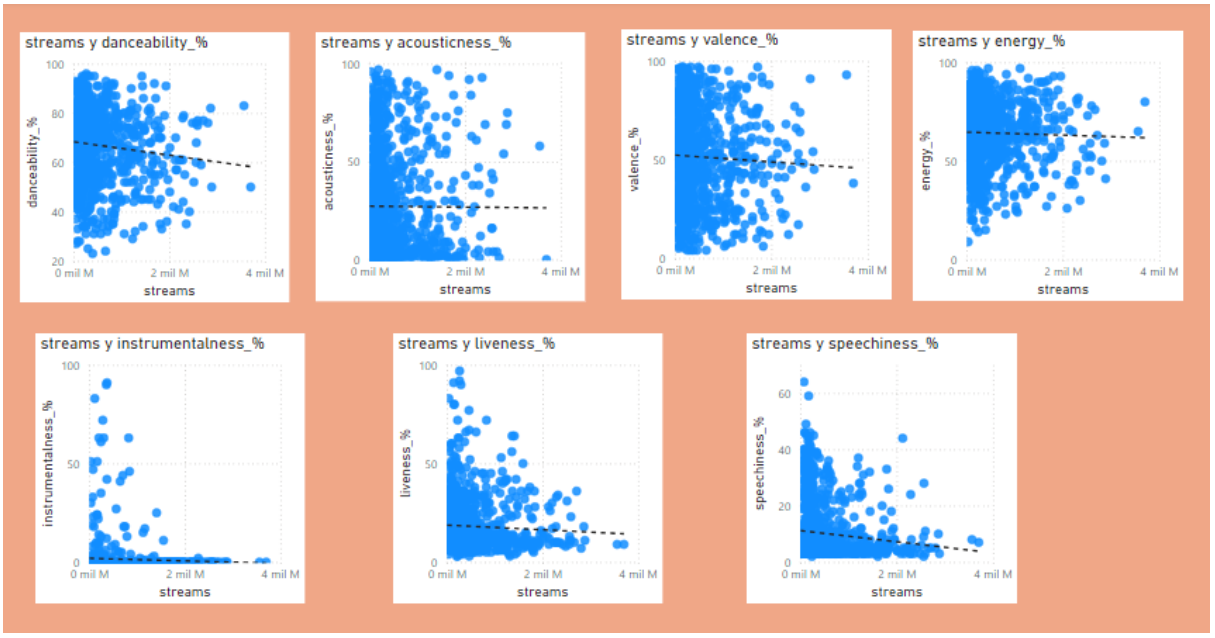
Hipótesis 5

Las características de la canción influyen en el éxito en términos de streams en Spotify.

Correlaciones negativas:

Estas variables no tienen relaciones entre sí, por lo que no influyen en el éxito de una canción por la cantidad de streams que tenga.

correlation_value_str	correlation_value_str	correlation_value_str	correlation_value_str	correlation_value_str	correlation_value_str	correlation_value_str
-0.10556527471...	-0.04163248151...	-0.02592698014...	-0.00499951659...	-0.04416555442...	-0.04947345948...	-0.11275108357...



- **Recomendaciones**

Presencia en más playlists de terceros pude ayudar a llegar a más público.

Participar en eventos musicales en vivo, llama mucho el interés de las personas.

Crear un press kit para que todas tus comunicaciones sintonicen con la misma partitura e introducir palabras clave que puedan ser tecleadas por el público objetivo.

- **Limitaciones/Próximos Pasos**

En un inicio me faltaba espacio para descargar Power BI, pero luego instale la opción más liviana.

Cuando instale Python no se me visualizaba los histogramas pero luego con ayuda de las compañeras y coaches pude solucionarlo.

El Próximo paso que tenemos en mente es más adelante ver hitos 2 y 3 para comprender mas a profundidad las hipótesis.

- **Enlaces de interés**

Carpeta de Proyecto 2 Hipótesis

https://drive.google.com/drive/folders/1JEBp6zYVQcAksLfj3o2yYrgIPNbQOEnA

Link para ver la ficha técnica en Notion

https://www.notion.so/Ficha-t-cnica-proyecto-2-Hip-tesis-4b80dc82d3a34c5e98ee58cf00a2f0a3?pvs=4