



Spotify / YouTube Data Analysis

Lisa Yu | Chenxi Ye | Nik King Fredel
Apr 30, 2025

OVERVIEW

Spotify and YouTube are two of the most influential platforms in the modern music ecosystem. Spotify provides streaming-based insights into user listening behavior, while YouTube captures video-based engagement metrics such as views, likes, and comments. The two platforms provide complementary perspectives in measuring a track's popularity and breakout potential. By analyzing data from both platforms, we hope to gain a deeper understanding of commercially successful songs and provide insights for studios looking to create hits.

GOALS

This project aims to help music studios make data-driven investment decisions by analyzing cross-platform music performance. Specifically, we utilize streaming data from Spotify and YouTube to identify:

- Tracks that are already high-performing, and
- Key audio features that may indicate potential breakout success.

By understanding which musical characteristics are associated with strong performance on different platforms, studios can better allocate resources to promote promising songs and guide future production choices.

DATASET OVERVIEW

The dataset used in this analysis includes over 20,000 tracks by more than 2,000 unique artists, combining basic track metadata, core audio features, and performance metrics from both Spotify and YouTube. Each track entry contains information such as artist name, album, track duration, Spotify stream count, and YouTube engagement data including views, likes, and comments. Audio features such as danceability, energy, loudness, and acousticness are also included to support quantitative comparison.

DELIVERABLES

Deliverables include:

- 1) Data Model
- 2) Data Preparation
- 3) Performance Analysis
- 4) Recommendations

[illegible]

To classify track performance, we applied NTILE(4) percentile ranking to both Spotify streams and YouTube views. Based on these rankings, tracks were grouped into three categories:

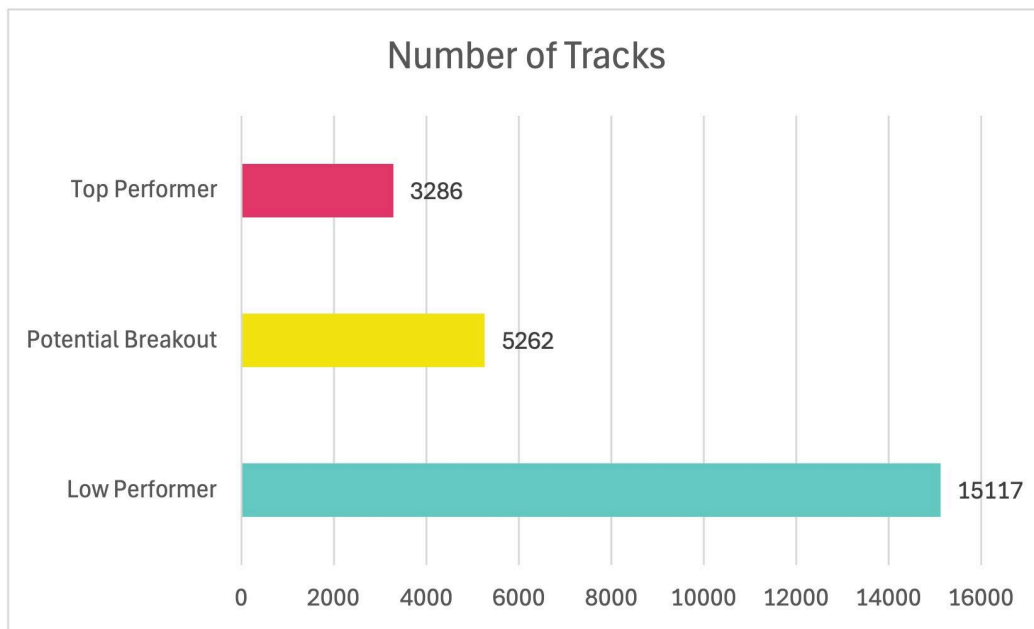
- All audio features were standardized using Z-score normalization to enable fair comparison across different scales.
- In later analysis, we further classified songs as Spotify-dominant or YouTube-dominant based on whether a track was in the top 25% of performance on one platform but not the other.

ANALYSIS

PERFORMANCE ANALYSIS 1: Clusters of Audio Features Common in Trending Tracks

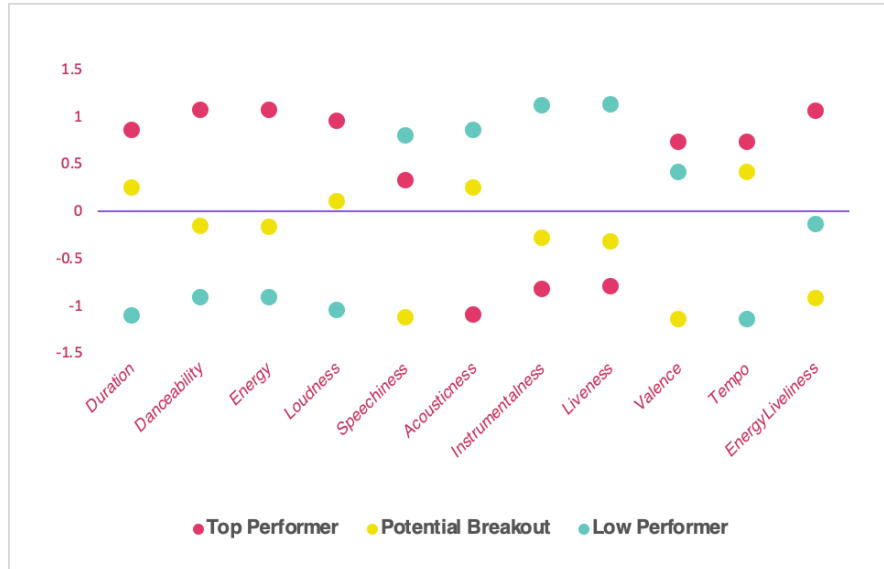
Number of Tracks in Quartile Investment Groups:

Investment Category	Number of Tracks
Low Performer	15117
Potential Breakout	5262
Top Performer	3286



Characteristics by Investment Group

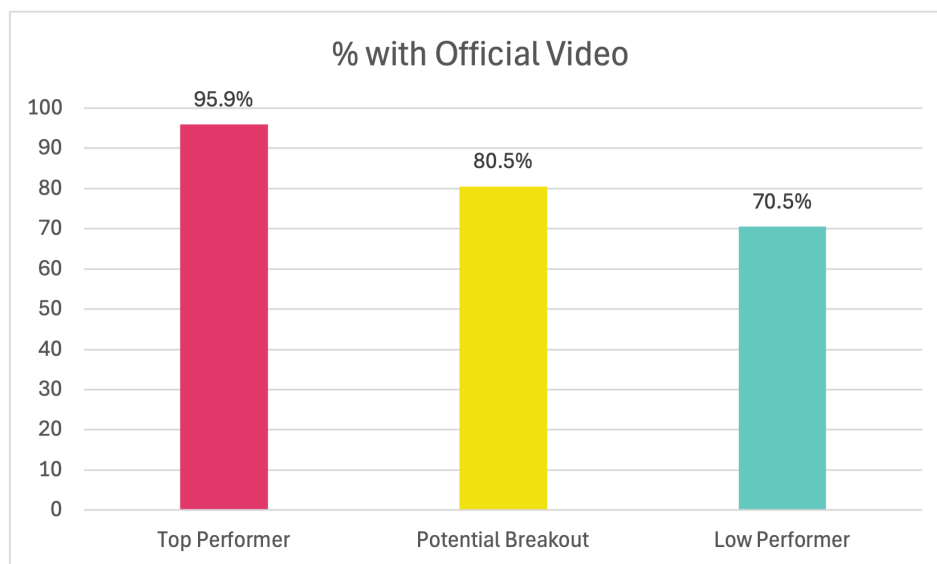
Investment Category	Duration (Min)	Danceability	Energy	Loudness	Speechiness	Instrumentalness	Liveness	Valence	Tempo	Energy Liveness
Top Performer	3.78	0.643	0.665	-6.3	0.089	0.022	0.18	0.52	121.55	5.496
Low Performer	3.65	0.606	0.617	-8.06	0.092	0.065	0.188	0.519	120.36	5.007
Potential Breakout	3.74	0.62	0.635	-7.05	0.08	0.034	0.182	0.514	121.35	5.2



PERFORMANCE ANALYSIS 2: Impact of Official Videos and Platforms

Official Video Impact

Investment Category	% with Official Video
Top Performer	95.9
Potential Breakout	80.5
Low Performer	70.5

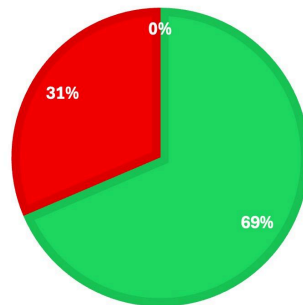


Dominant Platform Analysis

Category	Spotify	YouTube	Tie
Low Performer	82.7	17.1	0.1
Potential Breakout	55.4	44.6	0
Top Performer	68.6	31.4	0

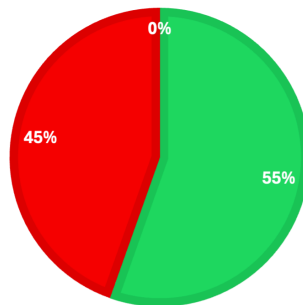
TOP PERFORMER

■ Spotify ■ YouTube ■ Tie



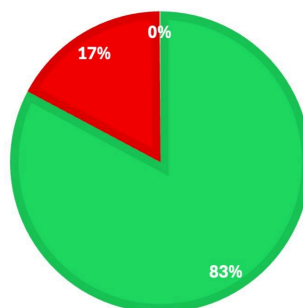
POTENTIAL BREAKOUT

■ Spotify ■ YouTube ■ Tie



LOW PERFORMER

■ Spotify ■ YouTube ■ Tie



PERFORMANCE ANALYSIS PART 3: Platform Specific Features and Popular Examples

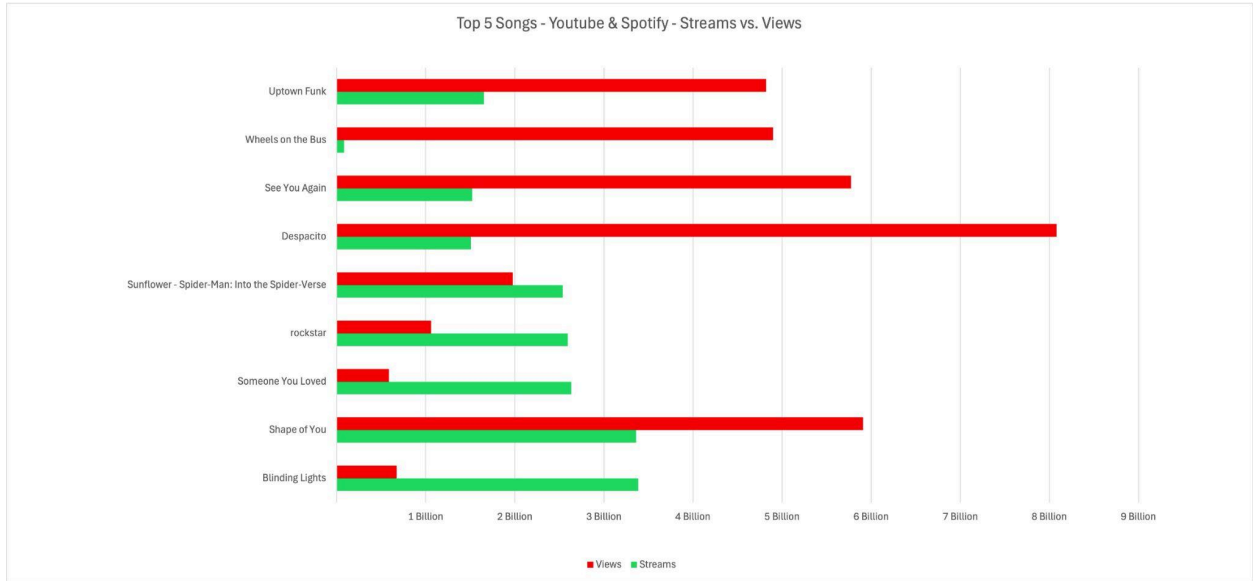
Audio Features

Platform	Track counts	Danceability	Energy	Speechiness	Instrumentalness	Liveness	Valence	Tempo	Duration (Min)
YouTube-Dominant	4292	0.63	0.653	0.082	0.53	0.28	0.026	121.94	3.85
Spotify-Dominant	4581	0.61	0.617	0.079	0.498	0.285	0.041	120.8	3.63

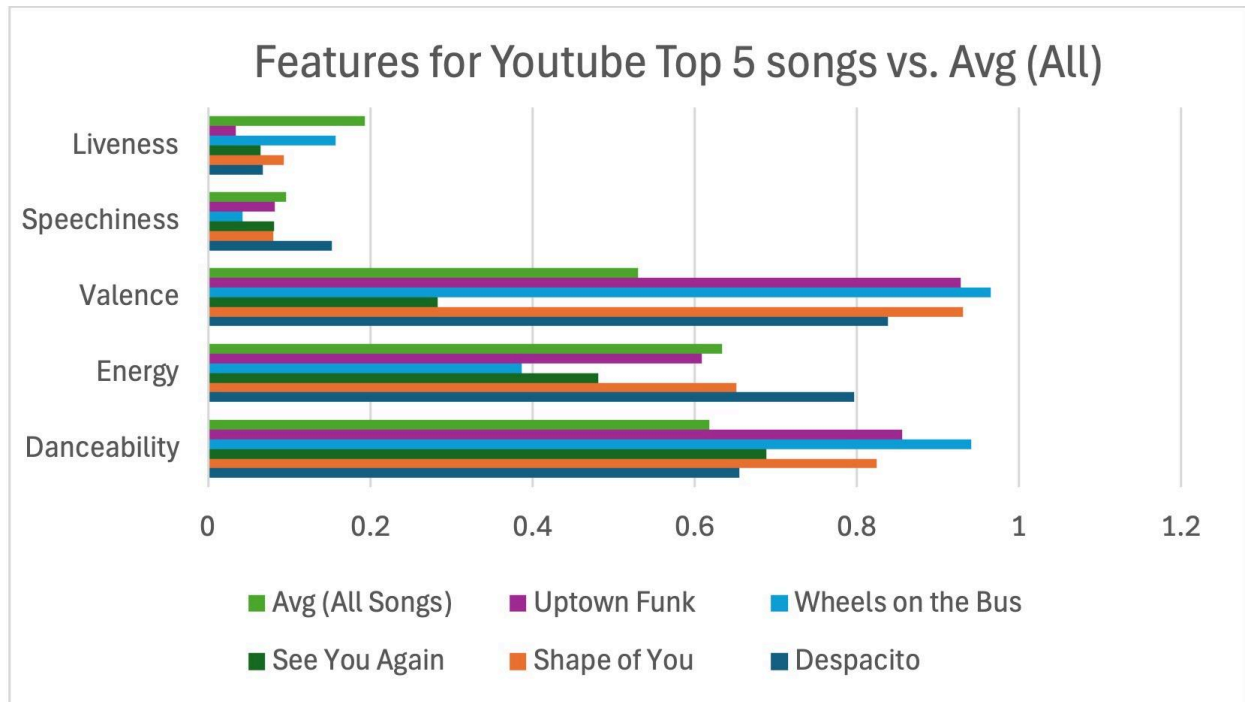
- Spotify-dominant songs tend to have higher values in acousticness and instrumentalness, indicating a preference for quieter, non-vocal, and musically rich tracks.
- YouTube-dominant songs, on the other hand, score higher in loudness, energy, speechiness, liveness, valence, and tempo. This suggests that tracks with more intensity, emotional expression, and social or vocal presence perform better on YouTube.

Top 5 Songs on Spotify and YouTube:

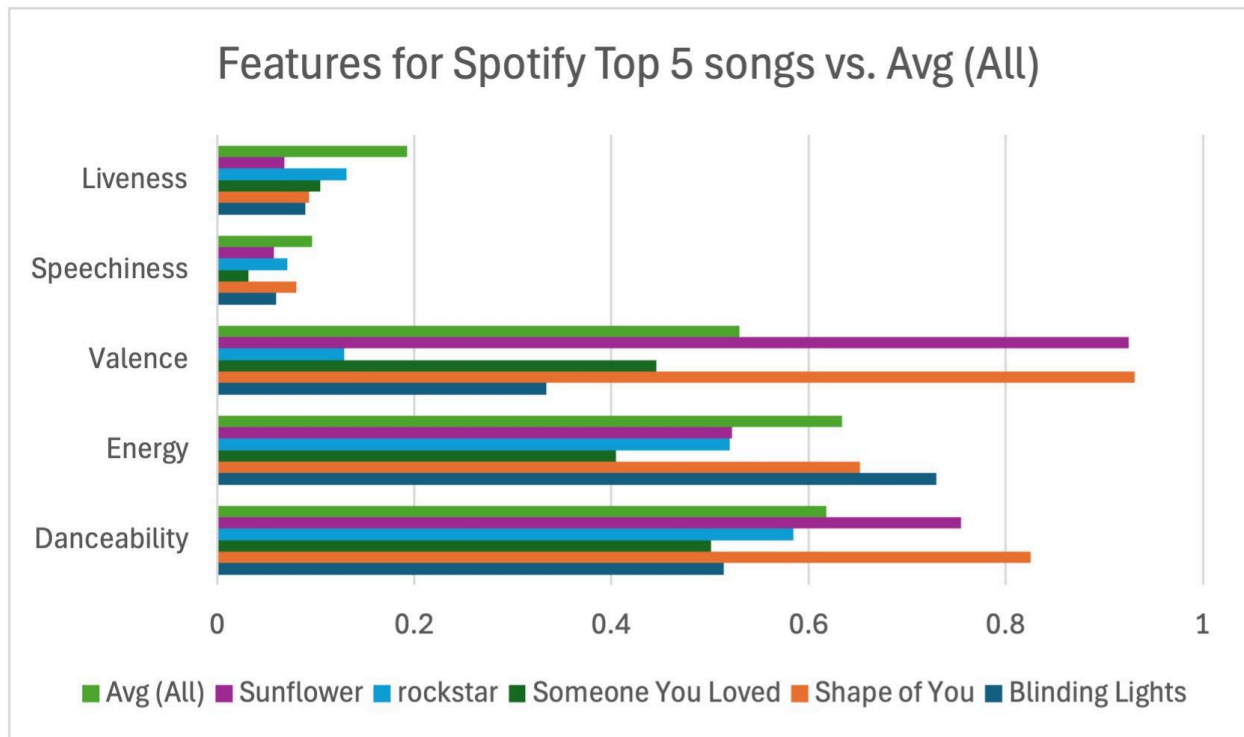
Track Name	Artist	Streams	Views
Blinking Lights	The Weeknd	3386520288	674164500
Shape of You	Ed Sheeran	3362005201	5908398479
Someone You Loved	Lewis Capaldi	2634013335	586768373
rockstar	Post Malone	2594926619	1060220169
Sunflower - Spider-Man: Into the Spider-Verse	Swae Lee	2538329799	1977389027
Despacito	Daddy Yankee & Luis Fonsi	1506598267	8079649362
See You Again	Charlie Puth & Wiz Khalifa	1521254554	5773798407
Wheels on the Bus	CoComelon	83434355	4898831101
Uptown Funk	Mark Ronson	1653819736	4821016218



Feature	Despacito	Shape of You	See You Again	Wheels on the Bus	Uptown Funk	Avg (All Songs)
Danceability	0.655	0.825	0.689	0.941	0.856	0.618
Energy	0.797	0.652	0.481	0.387	0.609	0.634
Valence	0.839	0.931	0.283	0.965	0.928	0.53
Speechiness	0.153	0.0802	0.0815	0.0427	0.0824	0.096
Liveness	0.067	0.0931	0.0649	0.157	0.0344	0.193



Feature	Blinding Lights	Shape of You	Someone You Loved	rockstar	Sunflower	Avg (All)
Danceability	0.514	0.825	0.501	0.585	0.755	0.618
Energy	0.73	0.652	0.405	0.52	0.522	0.634
Valence	0.334	0.931	0.446	0.129	0.925	0.53
Speechiness	0.0598	0.0802	0.0319	0.0712	0.0575	0.096
Liveness	0.0897	0.0931	0.105	0.131	0.0685	0.193



Top 10 Youtube Channels

Channel	Views
T-Series	47961585483
LuisFonsiVEVO	21702521043
CalvinHarrisVEVO	18281142527
Macklemore	17003410965
Ed Sheeran	16918224412
Hear This Music	15088505405
Cocomelon - Nursery Rhymes	14601671211
Bruno Mars	13013566838
KatyPerryVEVO	12476802428
DJSnakeVEVO	12455097762

RECOMMENDATIONS

Music Attributes

Based on our analysis of over 20,000 tracks, high-performing songs share distinct audio characteristics. We recommend that music studios prioritize tracks with the following features:

- High Danceability (> 0.62): Tracks with engaging rhythm are strongly associated with success.
- High Energy (> 0.635) and Loudness (> 7.05): Top-performing tracks tend to be intense and professionally produced.
- Faster Tempo (> 121.35 BPM): A slightly faster pace appears to support listener engagement.
- Low Acousticness (< 0.089), Instrumentalness (< 0.034), and Speechiness (< 0.09): Vocal-driven, fully produced tracks consistently outperform more acoustic or speech-heavy songs.

These traits can be used as a first-pass filter to evaluate the breakout potential of unreleased tracks.

Platform Strategy

Different platforms support different forms of success. Our findings suggest the following release and promotion strategy:

- Prioritize Spotify for long-term streaming engagement. Spotify is the stronger platform overall, particularly for tracks designed for passive or playlist-based listening.
- Leverage YouTube as a promotional springboard. High-performing YouTube tracks are typically energetic, visually engaging, and emotionally expressive. If a track performs well on YouTube but not yet on Spotify, it may be a “potential breakout” worth amplifying.
- Watch for dominant platform signals: A track trending more on YouTube than Spotify should be flagged for follow-up strategy and potential cross-platform promotion.

Long-Term Opportunity

Based on our analysis, we recommend that music studios consider building a predictive scoring system to evaluate investment-worthiness. Based on our analysis, this system could incorporate:

- Standardized audio features (such as danceability, energy, tempo, etc.)
- Basic platform indicators (e.g., dominant platform, official video presence)

This would allow studios to streamline decision-making and consistently identify high-potential songs early in the release cycle.

Limitations

- No genre classification: Our analysis did not control for genre, which may influence trends in energy, tempo, and platform performance.
- No visual analysis: For YouTube, we did not evaluate video content or production, which may impact user engagement and views.
- Timeframe not considered: The dataset does not specify release dates, so long-term popularity vs. viral trends could not be distinguished.
- No user behavior data: We focused on aggregated views and streams, not on listener intent (e.g., skip rate, replay rate).
- Limited platform scope: The analysis was based solely on Spotify and YouTube, excluding other relevant platforms like TikTok or Apple Music.

APPENDIX WITH SQL QUERIES

–Number of Tracks in each quartile (Slide 8)

```
CREATE VIEW v_tracks_per_quartile AS
WITH ranked_tracks AS (
  SELECT
    trk.track_pk,
    NTILE(4) OVER (ORDER BY sp.streams DESC) AS stream_quartile,
    NTILE(4) OVER (ORDER BY yt.views DESC) AS view_quartile
  FROM NKF_SPOT_TRACK trk
  LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_VID ytv ON ytv.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.track_fk = trk.track_pk
),
track_perf AS (
  SELECT
    track_pk,
    CASE
      WHEN stream_quartile = 1 AND view_quartile = 1 THEN 'Top Performer'
      WHEN stream_quartile = 1 OR view_quartile = 1 THEN 'Potential Breakout'
      ELSE 'Low Performer'
    END AS investment_category
  FROM ranked_tracks
)
SELECT
  investment_category,
  COUNT(*) AS num_tracks
FROM track_perf
GROUP BY investment_category
ORDER BY investment_category;
SELECT
  track,
  COUNT(DISTINCT title) AS num_videos
FROM NKF_SPOT_STAGE
GROUP BY track
HAVING COUNT(DISTINCT title) > 1;
```

–View for Audio Feature Averages by Investment Category

```
CREATE VIEW v_audio_features_by_category AS
WITH ranked_tracks AS (
  SELECT
    trk.track_pk,
    sp.streams,
    yt.views,
    NTILE(4) OVER (ORDER BY sp.streams DESC) AS stream_quartile,
    NTILE(4) OVER (ORDER BY yt.views DESC) AS view_quartile
  FROM NKF_SPOT_TRACK trk
  LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_VID ytv ON ytv.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.track_fk = trk.track_pk
),
track_perf AS (
  SELECT
    track_pk,
    CASE
      WHEN stream_quartile = 1 AND view_quartile = 1 THEN 'Top Performer'
      WHEN stream_quartile = 1 OR view_quartile = 1 THEN 'Potential Breakout'
      ELSE 'Low Performer'
    END AS investment_category
  FROM ranked_tracks
),
track_feats AS (
  SELECT
    track_pk,
    duration,
    danceability,
    energy,
    loudness,
    speechiness,
    instrumentalness,
    liveness,
    valence,
    tempo,
    energy_liveness
  FROM NKF_SPOT_FEATURE
)
```

```

SELECT
  tp.investment_category,
  ROUND(AVG(f.duration), 2) AS avg_duration_min,
  ROUND(AVG(f.danceability), 3) AS avg_danceability,
  ROUND(AVG(f.energy), 3) AS avg_energy,
  ROUND(AVG(f.loudness), 2) AS avg_loudness,
  ROUND(AVG(f.speechiness), 3) AS avg_speechiness,
  ROUND(AVG(f.instrumentalness), 3) AS avg_instrumentalness,
  ROUND(AVG(f.liveness), 3) AS avg_liveness,
  ROUND(AVG(f.valence), 3) AS avg_valence,
  ROUND(AVG(f.tempo), 2) AS avg_tempo,
  ROUND(AVG(f.energy_liveness), 3) AS avg_energy_liveness
FROM track_perf tp
JOIN track_feats f ON tp.track_pk = f.track_fk
GROUP BY tp.investment_category;

```

–Official Video Impact

```

CREATE VIEW v_official_video_impact AS
WITH ranked_tracks AS (
  SELECT
    trk.track_pk,
    ytvid.official_video,
    sp.streams,
    yt.views,
    NTILE(4) OVER (ORDER BY sp.streams DESC) AS stream_quartile,
    NTILE(4) OVER (ORDER BY yt.views DESC) AS view_quartile
  FROM NKF_SPOT_TRACK trk
  LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
  LEFT JOIN NKF_SPOT_YT_VID ytvid ON ytvid.vid_pk = ytv.yt_vid_fk
  LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.yt_vid_fk = ytvid.vid_pk
),
track_perf AS (
  SELECT
    track_pk,
    official_video,
    CASE
      WHEN stream_quartile = 1 AND view_quartile = 1 THEN 'Top Performer'

```

```

        WHEN stream_quartile = 1 OR view_quartile = 1 THEN 'Potential Breakout'
        ELSE 'Low Performer'
    END AS investment_category
FROM ranked_tracks
WHERE official_video IS NOT NULL
)
SELECT
    investment_category,
    COUNT(*) AS total_tracks,
    SUM(CASE WHEN official_video = 1 THEN 1 ELSE 0 END) AS tracks_with_official_video,
    ROUND(100.0 * SUM(CASE WHEN official_video = 1 THEN 1 ELSE 0 END) / COUNT(*),
1) AS pct_with_official_video
FROM track_perf
GROUP BY investment_category
ORDER BY investment_category;

```

–Dominant Platform Analysis

```

CREATE VIEW v_dominant_platform AS
WITH ranked_tracks AS (
    SELECT
        trk.track_pk,
        sp.streams,
        yt.views,
        NTILE(4) OVER (ORDER BY sp.streams DESC) AS stream_quartile,
        NTILE(4) OVER (ORDER BY yt.views DESC) AS view_quartile
    FROM NKF_SPOT_TRACK trk
    LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
    LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
    LEFT JOIN NKF_SPOT_YT_VID ytv ON ytv.vid_pk = ytv.yt_vid_fk
    LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.yt_vid_fk = ytv.vid_pk
),
track_perf AS (
    SELECT
        track_pk,
        streams,
        views,
        CASE

```



```

        WHEN stream_quartile = 1 AND view_quartile = 1 THEN 'Top Performer'
        WHEN stream_quartile = 1 OR view_quartile = 1 THEN 'Potential Breakout'
        ELSE 'Low Performer'
    END AS investment_category
FROM ranked_tracks
),
platform_dominance AS (
    SELECT
        tp.investment_category,
        tp.track_pk,
        tp.streams,
        tp.views,
        (tp.streams + tp.views) AS total_engagement,
        CASE
            WHEN tp.views > tp.streams THEN 'YouTube Dominated'
            WHEN tp.streams > tp.views THEN 'Spotify Dominated'
            ELSE 'Tie'
        END AS dominant_platform
    FROM track_perf tp
    WHERE tp.streams IS NOT NULL AND tp.views IS NOT NULL
)
SELECT
    investment_category,
    dominant_platform,
    COUNT(*) AS num_tracks,
    ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (PARTITION BY
investment_category), 1) AS pct_within_category
FROM platform_dominance
GROUP BY investment_category, dominant_platform
ORDER BY investment_category, dominant_platform;

```

–Feature Split

```

WITH track_stats AS (
    SELECT
        sp.TRACK_FK,
        sp.STREAMS,
        yt.VIEWS,
        NTILE(4) OVER (ORDER BY sp.STREAMS) AS streams_ntile,

```

```

    NTILE(4) OVER (ORDER BY yt.VIEWS) AS views_ntile
FROM SPOT_SP_STATS sp
JOIN SPOT_VID vid ON sp.TRACK_FK = vid.TRACK_FK
JOIN SPOT_YT_STATS yt ON vid.VID_PK = yt.VID_FK
),
dominant_platform AS (
    SELECT
        ts.TRACK_FK,
        ts.STREAMS,
        ts.VIEWS,
        CASE
            WHEN ts.streams_ntile = 4 AND ts.views_ntile < 4 THEN 'Spotify-Dominant'
            WHEN ts.streams_ntile < 4 AND ts.views_ntile = 4 THEN 'YouTube-Dominant'
            ELSE NULL
        END AS platform_category
    FROM track_stats ts
    WHERE ts.streams_ntile = 4 OR ts.views_ntile = 4
),
joined_features AS (
    SELECT
        dp.platform_category,
        f.DANCEABILITY,
        f.ENERGY,
        f.SPEECHINESS,
        f.VALENCE,
        f.ACOUSTICNESS,
        f.INSTRUMENTALNESS,
        f.TEMPO,
        f.DURATION_MIN
    FROM dominant_platform dp
    JOIN SPOT_TRACK t ON dp.TRACK_FK = t.TRACK_PK
    JOIN SPOT_FEATURE f ON t.TRACK_PK = f.TRACK_FK
    WHERE dp.platform_category IS NOT NULL
)
SELECT
    platform_category,
    COUNT(*) AS track_count,
    ROUND(AVG(DANCEABILITY), 3) AS avg_danceability,
    ROUND(AVG(ENERGY), 3) AS avg_energy,
    ROUND(AVG(SPEECHINESS), 3) AS avg_speechiness,

```

```

ROUND(AVG(VALENCE), 3) AS avg_valence,
ROUND(AVG(ACOUSTICNESS), 3) AS avg_acousticness,
ROUND(AVG(INSTRUMENTALNESS), 3) AS avg_instrumentalness,
ROUND(AVG(TEMPO), 2) AS avg_tempo,
ROUND(AVG(DURATION_MIN), 2) AS avg_duration_min
FROM joined_features
GROUP BY platform_category;

```

–Top 5 YT Video

```

CREATE VIEW v_top_youtube_tracks AS
SELECT
    trk.track_name,
    art.artist_name,
    yt.views,
    sp.streams,
    f.danceability,
    f.energy,
    f.valence,
    f.speechiness,
    f.instrumentalness,
    f.liveness,
    f.loudness,
    f.tempo,
    f.duration
FROM NKF_SPOT_TRACK trk
JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
JOIN NKF_SPOT_YT_VID ytv ON ytv.vid_pk = ytv.yt_vid_fk
JOIN NKF_SPOT_YT_STAT yt ON yt.yt_vid_fk = ytv.vid_pk
LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
JOIN NKF_SPOT_TRACK_ARTIST ta ON ta.track_fk = trk.track_pk
JOIN NKF_SPOT_ARTIST art ON art.artist_pk = ta.artist_fk
LEFT JOIN NKF_SPOT_FEATURE f ON f.track_fk = trk.track_pk
ORDER BY yt.views DESC;

```

–Top 5 Spotify Tracks

```

CREATE VIEW v_top_spotify_tracks AS

```

```

SELECT
    trk.track_name,
    art.artist_name,
    sp.streams,
    yt.views,
    f.danceability,
    f.energy,
    f.valence,
    f.speechiness,
    f.instrumentalness,
    f.liveness,
    f.loudness,
    f.tempo,
    f.duration
FROM NKF_SPOT_TRACK trk
JOIN NKF_SPOT_SP_STAT sp ON trk.track_pk = sp.track_fk
LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
LEFT JOIN NKF_SPOT_YT_VID ytv ON ytv.track_fk = trk.track_pk
LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.track_fk = trk.track_pk
JOIN NKF_SPOT_TRACK_ARTIST ta ON ta.track_fk = trk.track_pk
JOIN NKF_SPOT_ARTIST art ON art.artist_pk = ta.artist_fk
LEFT JOIN NKF_SPOT_FEATURE f ON f.track_fk = trk.track_pk
ORDER BY sp.streams DESC;

```

–View for Top YouTube Channels by Number of Top Performer Tracks

```

CREATE VIEW v_top_channels AS
WITH ranked_tracks AS (
    SELECT
        trk.track_pk,
        ch.channel_name,
        sp.streams,
        yt.views,
        NTILE(4) OVER (ORDER BY sp.streams DESC) AS stream_quartile,
        NTILE(4) OVER (ORDER BY yt.views DESC) AS view_quartile
    FROM NKF_SPOT_TRACK trk
    LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
    LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk

```

```

LEFT JOIN NKF_SPOT_YT_VID ytvid ON ytvid.vid_pk = ytv.t.yt_vid_fk
LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.yt_vid_fk = ytvid.vid_pk
LEFT JOIN NKF_SPOT_YT_VID_YT_CHANNEL vc ON vc.vid_fk = ytvid.vid_pk
LEFT JOIN NKF_SPOT_YT_CHANNEL ch ON ch.channel_pk = vc.channel_fk
),
track_perf AS (
  SELECT
    track_pk,
    channel_name,
    CASE
      WHEN stream_quartile = 1 AND view_quartile = 1 THEN 'Top Performer'
      WHEN stream_quartile = 1 OR view_quartile = 1 THEN 'Potential Breakout'
      ELSE 'Low Performer'
    END AS investment_category
  FROM ranked_tracks
)
SELECT
  tp.channel_name,
  COUNT(*) AS top_performer_count
FROM track_perf tp
WHERE tp.investment_category = 'Top Performer'
GROUP BY tp.channel_name
ORDER BY top_performer_count DESC;

```

–Streams vs Views on Top 5 Spotify / YT

```

CREATE VIEW v_streams_vs_views_all AS
SELECT
  trk.track_name,
  art.artist_name,
  sp.streams,
  yt.views
FROM NKF_SPOT_TRACK trk
LEFT JOIN NKF_SPOT_SP_STAT sp ON sp.track_fk = trk.track_pk
LEFT JOIN NKF_SPOT_YT_VID_TRACK ytv ON ytv.track_fk = trk.track_pk
LEFT JOIN NKF_SPOT_YT_VID ytvid ON ytvid.vid_pk = ytv.yt_vid_fk
LEFT JOIN NKF_SPOT_YT_STAT yt ON yt.yt_vid_fk = ytvid.vid_pk
JOIN NKF_SPOT_TRACK_ARTIST ta ON ta.track_fk = trk.track_pk

```

```
JOIN NKF_SPOT_ARTIST art ON art.artist_pk = ta.artist_fk;
```

```
SELECT * FROM v_streams_vs_views_all  
ORDER BY sp.streams DESC;
```

```
SELECT * FROM v_streams_vs_views_all  
ORDER BY yt.views DESC;
```