

## Varying the Time Window:

- **1-minute Window:**
  - "Now Trending" results will update very frequently
  - Latency is very low, as new changes are being detected almost instantaneously.
  - Due to low latency this could lead to a less stable "Now Trending" list.
- **10-minute Window:**
  - "Now Trending" results will update less frequently, reflecting a longer period of activity.
  - Latency is higher, as it takes longer for the list to be updated.
  - The list is more stable.
  - This provides a more reliable "Now Trending" list, but it might not be up to date all the time.
- **Trade-offs:**
  - **Latency vs. Stability:** Shorter windows provide lower latency but also less stability, while longer windows offer higher stability but also increased latency.

```
lab5 — spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.5 — spark-submit — 530:533 — java -cp ~/Desk...
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|101|4|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|202|2|2|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|303|2|3|
+-----+-----+-----+-----+
=== Batch: 2 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|APAC|101|5|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|APAC|202|2|2|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|APAC|505|1|3|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|EU|303|2|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|EU|505|2|2|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|EU|202|1|3|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|101|5|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|303|3|2|
+-----+-----+-----+-----+
=== Batch: 3 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|404|3|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|202|1|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|303|1|3|
+-----+-----+-----+-----+
=== Batch: 4 ===
```

Now trending list with 1 minute window

```
lab5 -- spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.5 -- spark-submit -- 487:554 - java -cp ~/Desktop/ai601-data-engineering/venv/lib/python3.10/...

+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|APAC|101|5|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|APAC|202|5|2|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|APAC|505|1|3|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|EU|303|2|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|EU|505|2|2|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|EU|202|1|3|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|101|5|1|
|{2025-03-17 00:27:00, 2025-03-17 00:28:00}|US|303|3|2|
+-----+-----+-----+-----+

=== Batch: 3 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|404|3|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|202|1|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|303|1|3|
+-----+-----+-----+-----+

=== Batch: 4 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|101|1|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|202|1|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|404|1|3|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|303|2|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|101|1|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|404|2|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|505|2|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|303|1|3|
+-----+-----+-----+-----+

=== Batch: 5 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|505|1|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|404|4|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|101|1|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|505|3|3|
+-----+-----+-----+-----+

=== Batch: 6 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|303|2|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|101|2|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|US|202|2|3|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|202|2|1|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|EU|101|1|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|505|4|2|
|{2025-03-17 00:28:00, 2025-03-17 00:29:00}|APAC|202|1|3|
+-----+-----+-----+-----+
```

Now trending list with 1 minute window

```
lab5 -- spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.5 -- spark-submit -- python3 - java -cp ~/Desk...

+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|US|505|5|1|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|US|101|4|2|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|US|303|3|3|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|APAC|505|3|1|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|APAC|202|3|2|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|APAC|101|1|3|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|EU|303|8|1|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|EU|505|1|2|
+-----+-----+-----+-----+

=== Batch: 5 ===
+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|US|101|7|1|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|US|303|4|2|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|APAC|505|4|1|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|EU|101|6|1|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|EU|404|3|2|
|{2025-03-17 00:20:00, 2025-03-17 00:30:00}|EU|505|2|3|
+-----+-----+-----+-----+

=== Batch: 6 ===
```

Now trending list with 10 minute window

## Add Skip/Like Actions:

- The provided code now includes "like" actions in the aggregation.
- We also calculate the skip ratio now.

```
lab5 — python music_producer.py — python — python music_producer.p...
(.venv) → lab5 git:(main) ✖ python music_producer.py

Sent event: {'song_id': 505, 'timestamp': 1742153477.152459, 'region': 'EU', 'action': 'skip'}
Sent event: {'song_id': 202, 'timestamp': 1742153478.719263, 'region': 'US', 'action': 'like'}
Sent event: {'song_id': 202, 'timestamp': 1742153479.6155488, 'region': 'APAC', 'action': 'play'}
Sent event: {'song_id': 404, 'timestamp': 1742153481.512696, 'region': 'US', 'action': 'play'}
Sent event: {'song_id': 505, 'timestamp': 1742153483.2298298, 'region': 'US', 'action': 'play'}
Sent event: {'song_id': 505, 'timestamp': 1742153484.931301, 'region': 'EU', 'action': 'play'}
Sent event: {'song_id': 303, 'timestamp': 1742153485.9456398, 'region': 'EU', 'action': 'like'}
Sent event: {'song_id': 101, 'timestamp': 1742153487.781919, 'region': 'APAC', 'action': 'play'}
Sent event: {'song_id': 202, 'timestamp': 1742153489.494802, 'region': 'APAC', 'action': 'skip'}
```

window		region	song_id	play_count	skip_count	skip_ratio
{2025-03-17 00:35:00, 2025-03-17 00:36:00}		EU	505	1	2	0.6666666666666666
{2025-03-17 00:35:00, 2025-03-17 00:36:00}		APAC	404	1	3	0.75
{2025-03-17 00:35:00, 2025-03-17 00:36:00}		US	101	0	1	1.0
{2025-03-17 00:35:00, 2025-03-17 00:36:00}		EU	404	0	4	1.0
{2025-03-17 00:35:00, 2025-03-17 00:36:00}		APAC	303	0	2	1.0

## Skip Ratio and Skip/Like Actions Implementation



## Set a Different Micro-Batch Interval:

- **1-second Trigger:**
  - PySpark processes new data every second, trying to provide very frequent updates.
  - CPU usage will be higher, as Spark is constantly checking for and processing new data.
  - Results appear faster.
- **10-15 second Trigger:**
  - PySpark tries to process data every 10-15 seconds based on the trigger.
  - CPU usage will be lower than a 1 second trigger.
  - Results will appear every 10-15 seconds depending on how faster it can process data.
- **Performance vs. Overhead:**
  - Shorter trigger intervals provide lower latency and provide faster updates but increase CPU overhead.
  - Longer intervals reduce CPU usage but introduce more latency.
  - The optimal choice depends on the application's needs. Real-time updates may require a shorter interval despite the overhead, while resource efficiency favors a longer one. We can decide on the tradeoff based on the task at hand.

```
No data in this batch.
25/03/17 00:50:45 WARN ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 15000 milliseconds, but spent 18152 milliseconds
=== Batch: 1 ===
+-----+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+-----+
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|US|101|1|1|
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|US|404|1|2|
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|EU|303|1|1|
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|EU|505|1|2|
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|APAC|202|1|1|
+-----+-----+-----+-----+-----+

=== Batch: 2 ===
+-----+-----+-----+-----+-----+
|window|region|song_id|count|rn|
+-----+-----+-----+-----+-----+
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|US|404|2|1|
|{2025-03-17 00:50:00, 2025-03-17 00:52:00}|US|202|1|2|
+-----+-----+-----+-----+-----+

=== Batch: 3 ===
[]

No data in this batch.
25/03/17 00:45:44 WARN ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 1000 milliseconds, but spent 18759 milliseconds
=== Batch: 1 ===
Top 3 Songs:
+-----+-----+-----+-----+-----+-----+
|window|region|song_id|play_count|skip_count|skip_ratio|rn|
+-----+-----+-----+-----+-----+-----+
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|EU|505|1|0|0.0|1|
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|US|404|0|0|0.0|1|
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|US|101|0|1|1.0|2|
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|US|202|0|1|1.0|3|
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|APAC|202|2|0|0.0|1|
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|APAC|303|1|0|0.0|2|
|{2025-03-17 00:45:00, 2025-03-17 00:46:00}|APAC|404|0|0|0.0|3|
+-----+-----+-----+-----+-----+-----+
```

Trigger intervals of 1 second and 15 seconds.

\* Notice the warning that Current batch is falling behind because the window is much shorter than the processing time required.