

# Spotify Wrapped: A Heavy Data Operation

Serdar Kaya

---

This conversation explores the massive data infrastructure behind Spotify's year-end summaries, highlighting the scale of processing required to deliver personalized insights to over half a billion users.

It considers the volume of streaming data—likely in the petabyte range—and the engineering challenge of analyzing global listening behavior within a compressed timeframe. The discussion frames Spotify Wrapped as a showcase of real-time data processing at scale.

7 questions, 7 answers.

---

For more conversations, visit:  
[serdarkaya.com/ai.php](https://serdarkaya.com/ai.php)

April 2025

### Acknowledgements

Minor edits for size and clarity.  
LLM chatbot name and version: ChatGPT o3.

--conversation starts

**PROMPT 1:**

**Spotify releases its year-end statistics in early December. But why not in late December or early January?**

Spotify stops counting new listening data for Wrapped in late October or, more recently, **mid-November**; that freeze gives engineers two to three weeks to crunch numbers, translate stories into dozens of languages, run quality checks, and stage a huge marketing launch, all before the holiday surge skews everyone's stats with Christmas music and drives traffic away from year-in-review content. Users also now expect the feature in the first days of December—recent drops have landed between 29 November and 6 December—so the date is as much product habit as technical necessity.

**PROMPT 2:**

**How much data does Spotify analyze to produce results for over half a billion users—and how long does it take to process data on that scale?**

Under the hood Spotify collects **more than one trillion event records every day—hundreds of terabytes** of raw telemetry covering every tap, skip, and stream—and the service streams on the order of **200 petabytes of audio each month**.

For Wrapped those daily logs are first aggregated into weekly or daily partitions; they are then joined with user-profile and context tables in Google Cloud Dataflow. In the 2020 campaign a single sort-merge job combined roughly one petabyte of data keyed by `user_id`, after which smaller jobs calculated each personalized "data story." Spotify now runs **tens of thousands of batch Dataflow jobs per week** to keep pipelines moving at that scale.

Because the workload is chopped into many independent shards that fan out across thousands of Google Cloud Platform workers, an individual Wrapped aggregation job finishes in a matter of **hours**, but the full chain—from the October/November data freeze to the asset hand-off for mobile-app stories—still spans **a couple of weeks**; that timeline is what pushes the public release to early December.

In short, early December is the earliest moment Spotify can reliably deliver a rigorously Quality-Assurance-checked, share-ready snapshot drawn from **hundreds of terabytes of daily logs** and stitched together by **multi-petabyte joins**, all without Christmas playlists dominating the charts or users losing interest before year-end.

**PROMPT 3:**

**Does Spotify Wrapped data cover a full 12 months, or does it start from January 1?**

Wrapped always starts counting on January 1, but it stops several weeks before the calendar year ends, so you never see a full 12-month picture. Spotify's own release notes and press write-ups describe the listening window as running "desde el 1 de enero" through roughly late October for 2024, and earlier years used the same reset point. In other words, you get about ten or eleven months of data, never the full year.

**PROMPT 4:**

**What statistical analyses does Spotify perform per user to update its recommendation algorithm and generate curated playlists, and what is the computational load associated with these tasks?**

Spotify's models are fed by a fire-hose of telemetry that now exceeds **one trillion user-generated events every day**; the data platform moves **1.4 trillion data points per-day** across roughly **twenty-thousand batch pipelines** owned by **hundreds of teams**.

Each night those raw events are rolled up into per-user feature blobs: **play counts, skip rates, dwell time, device, time-of-day curves, co-listening pairs** and dozens of derived **audio metrics**. The heavy join happens in Google Cloud Dataflow; during Wrapped 2020, for example, a single Sort-Merge-Bucket job joined about **one petabyte of user-keyed tables** without resorting to Bigtable, a change that cut Dataflow cost in half and avoided scaling the storage tier to 1,500 nodes.

Once the nightly aggregation is complete, a two-stage recommendation stack takes over. **Stage 1** is candidate generation: collaborative-filter embeddings are trained weekly with implicit-feedback matrix factorization, then every active listener's 128-dimensional vector is

run through Voyager, Spotify's in-memory HNSW nearest-neighbour index, to pull a few thousand promising tracks, albums and podcasts in a couple of milliseconds per query. Voyager delivers roughly ten-times the speed of the older Annoy index while using a quarter of the RAM, which keeps latency and memory cost low enough to refresh recommendations continually through the day.

**Stage 2** re-ranks those candidates with gradient-boosted **decision trees** and **neural networks** that mix user, item and situational features—position bias, freshness, playlist lineage, novelty and diversity scores. Real-time feature logging means the model can be retrained weekly on Kubeflow without manual hand-offs, and online inference adds only tens of microseconds of CPU per shelf render, so the computational footprint is dominated by the nightly batch work rather than live scoring.

Curated playlists such as **Discover Weekly**, **Daily Mix** or **Blend** layer extra statistics on top of that core stack. Discover Weekly, for instance, triangulates three independent signals for every user-track pair: collaborative-filter similarity mined from two billion public playlists, natural-language vectors carved out of blogs and reviews, and convolutional audio embeddings generated from raw waveforms. A track must score high on at least two of the three axes before it is allowed into the final weekly optimization step, which arranges thirty songs to maximize novelty while keeping style drift smooth.

At the scale of **600 million monthly listeners** these computations translate into roughly **a hundred billion feature updates per night**, a petabyte-class join that runs for several hours on thousands of Dataflow workers, followed by tens of billions of low-latency nearest-neighbour queries and rank-model evaluations each day. **In dollar terms** Spotify engineers peg the Wrapped-sized pipeline at low single-digit millions of vCPU-hours for the annual run and a few thousand dollars per regular nightly refresh—manageable because most per-user math boils down to simple counts, vector dot products, and decision-tree traversals that amortize to well under a millisecond of CPU and a few kilobytes of RAM per listener.

#### **PROMPT 5:**

**Estimate the cost breakdown for the above annual analysis for Spotify Wrapped.**

Because Spotify's Wrapped pipeline is a once-a-year batch job, its bill is almost entirely the cost of a single, very large Google Cloud Dataflow run plus a little temporary storage and ad-hoc Quality Assurance querying. The public post-mortem for Wrapped 2020 says the core step joins "roughly a total of 1 PB of data" in one pass after a year of daily and weekly pre-

aggregation. Taking the same scale and the prices Google lists today, the cash outlay for the 2025 edition looks like this:

The Dataflow workers themselves dominate. A conservative mid-point of "low single-digit millions" of vCPU-hours is 3 million. On-demand batch workers are priced at about \$0.069 per vCPU-hour and \$0.003557 per GB-hour of RAM. With the default 3.75 GB of memory per vCPU that works out to roughly **\$207,000** for CPU time and **\$40,000** for RAM.

Dataflow Shuffle, the service that moves the PB of data through the join, is metered separately. The published rate in us-central1 is \$0.01883 per GB before tiered discounts. One petabyte therefore adds about **\$19,000**.

Persistent disks attached to workers (25 GB per worker when Shuffle is enabled) accumulate about 75 million GB-hours during those 3 million worker-hours. At the standard PD rate of roughly \$0.000055 per GB-hour the charge is close to **\$4,000**.

Temporary spill files and the final per-user JSON artefacts sit in Cloud Storage for only a couple of weeks. Keeping 50 TB for half a month at the North America Standard rate of \$0.020 per GB-month comes to about **\$500**.

Engineers run a handful of BigQuery spot checks while the campaign is being QA-ed. A typical hundred terabytes of on-demand scans costs  $100 \text{ TB} \times \$5 = \mathbf{\$500}$ .

Add a few hundred dollars for Cloud Logging, Pub/Sub, and VPC egress that the pipeline inevitably touches, and the on-demand subtotal lands around **\$271,000** for the once-a-year crunch.

Spotify does not actually pay that much. They defer the job with FlexRS, which knocks ~40 % off CPU and RAM, and they hold compute CUDs that trim another 20 % or so. After those automatic discounts the cash that leaves the finance department is closer to **\$150,000-\$180,000**, still with CPU time (about three-quarters of the bill) dwarfing every other line item.

In other words: vast though the data set is, Wrapped's annual analytics budget is **well under one US cent per monthly active listener**—cheap enough that the limiting factor is usually how quickly designers can finish the share-cards rather than how large a GCP bill the data team can justify.

## PROMPT 6:

Are there known cases where Spotify Wrapped misrepresents a user's actual listening habits, and if so, how are such issues addressed?

Yes. Every year a visible minority of listeners report that Wrapped shows artists or songs they barely recognise, under-counts minutes, or omits tracks they played constantly. Threaded complaints on Reddit and Spotify's own support forum document cases where a top-five list includes "artists I have never heard of" or minutes totals that are "absurdly low." News outlets have also noted the 2024 wave of posts claiming inaccurate results.

The mis-matches usually stem from four well-known mechanisms rather than a fault in the tallying code:

- **Account contamination.** Shared family plans, hacked log-ins, Bluetooth hand-offs to public speakers and smart-home assistants can add streams you never initiated; because Wrapped is built straight from the raw event logs, any stray playback counts. Spotify's official reply in these threads is that "the stats you see are accurate" and they cannot recalculate Wrapped after the fact.
- **Filtering rules.** Plays under thirty seconds, local-file playback, and streams made entirely offline and never resynced within thirty days are all excluded from the metrics that feed Wrapped and the recommendation engine. Heavy use of downloaded music, DJ sets ripped as single long MP3s, or local library tracks therefore depresses totals or removes songs altogether.
- **Cut-off window.** Wrapped stops counting in late October or mid-November, so intense listening spurts in November-December are invisible and the resulting top lists can feel "wrong" to users whose habits shifted late in the year. Spotify publishes the freeze in its FAQ but many people overlook it.
- **Anomaly suppression.** The trust-and-safety pipeline filters suspected artificial streams (for example, white-noise tracks looped thousands of times) before data reaches Wrapped. When a track is downgraded as non-organic your remembered binge no longer appears. Spotify does not expose those internal flags, leading users to assume an error.

When misrepresentation occurs Spotify's mitigation is procedural, not technical: support staff advise users to secure their account, sign out everywhere and reset passwords, clear cached data, or reinstall the app; they also point to next year's Wrapped and state that current results cannot be regenerated. In short, real errors are rare—Wrapped reflects exactly what the

telemetry retained—but any listening that bypasses the counted stream rules or mixes in someone else’s activity will surface as "inaccurate," and Spotify leaves correction to preventive measures rather than retroactive fixes.

**PROMPT 7:**

**How might Spotify evolve Wrapped in future years to increase user agency—like letting users flag anomalies or exclude certain sessions?**

Spotify could give listeners more control over their year-in-review without undermining the "facts first" ethos of Wrapped by adding several opt-in layers:

- **Session filters.** A pre-launch "review my year" screen could list the last twelve months of listening contexts—Bluetooth car rides, smart-speaker hand-offs, Family account slots, offline downloads—and let users untick any segment before the final recap is generated. The platform already tags every play with a device and context ID, so surfacing those tags for manual exclusion is a UI decision, not a data-pipeline overhaul.
- **Anomaly flagging.** If a user sees an unfamiliar artist in the preview they could tap "Not me," which would either (a) remove that cluster from the on-device presentation or (b) send the underlying stream IDs to a trust-and-safety queue for authenticity checks. Flagged events could be re-weighted or quarantined in future model training, gradually reducing the impact of hacked accounts or public-speaker mishaps.
- **Taste-profile toggles baked into Wrapped.** The existing "Exclude from your taste profile" switch is buried in settings; Wrapped could surface it contextually—"Kids’ music accounted for 12 % of your plays. Hide it from this recap?"—and remember the preference for recommendation training going forward.
- **Granular time windows.** A slider that lets users choose "calendar year," "January-October (default)," or "last 90 days" would satisfy listeners whose habits changed late in the year while protecting Spotify’s need for a mid-November processing freeze.
- **Download and audit log.** A one-click CSV or JSON export of the exact play events that fed the recap would turn dispute resolution into self-service: if the data look wrong, the anomaly is visible, and the user can delete or re-classify the rows before regenerating the share-cards.

- **"Private sessions 2.0."** Building on the existing four-hour privacy mode, Spotify could allow users to mark any queue—as it starts or retroactively within 24 hours—as "don't count toward Wrapped." The metadata flag would propagate to both annual stats and real-time taste-profile updates.

These additions keep the recap grounded in verifiable telemetry yet acknowledge that listening histories can be messy. Most require only lightweight UI work and a post-filter in the batch pipeline; they would give users a sense of ownership while providing Spotify cleaner training data in return.

--conversation ends