How Graphs, Trees, and Probability Decide Your Spotify Recommendations

Jordan Todev

March - April 2026

1 Introduction

Every week, Spotify hands me a playlist of about 30 songs I've never heard before, often in languages I don't speak or genres I've never tried. I love Metallica's heavy riffs, but it recommends Linkin Park's angst-filled rap-rock - and now it's my favourite too. Then Green Day's punk energy appears, followed by Blink-182's pop-punk hooks. Is this magic, or just maths hiding inside a giant graph?

In this essay, I'll show that Spotify's "magic" is built on three big ideas: graph theory connecting my Metallica to Blink-182, trees deciding playlist order, and probability predicting what I'll love next. By the end, we'll see how my listening history becomes perfect Discover Weekly recommendations through some seriously cool mathematics.

2 The hidden graph - songs, users, and features

At its core, Spotify builds massive graphs. Songs connect to songs, users connect to songs, even song features (energy, tempo, key) connect everything. My listening creates edges: I love Enter Sandman, metal fans who like it also love Numb, Linkin Park fans love American Idiot, punk fans love All the Small Things.

This makes a 4-step recommendation chain: Enter Sandman → Numb → American Idiot → All the Small Things

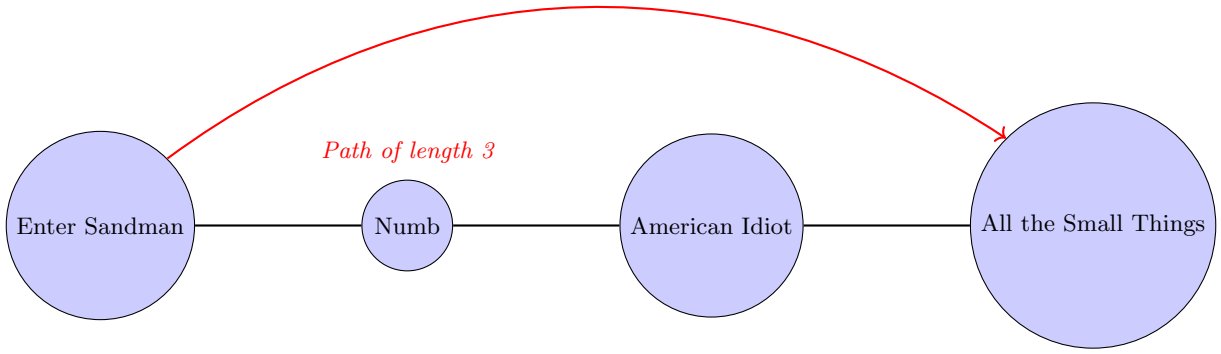

Figure 1: My Metallica → Blink-182 recommendation chain.

But there's more - users and songs make bipartite graphs (users left, songs right):

Formally: $S = \{s_1, s_2, \ldots, s_n\}$ songs, $U = \{u_1, u_2, \ldots, u_m\}$ users. Edge u_i-s_j if I listen to song j. Sparsity: $\frac{|E|}{m \cdot n} \approx 10^{-6}$.

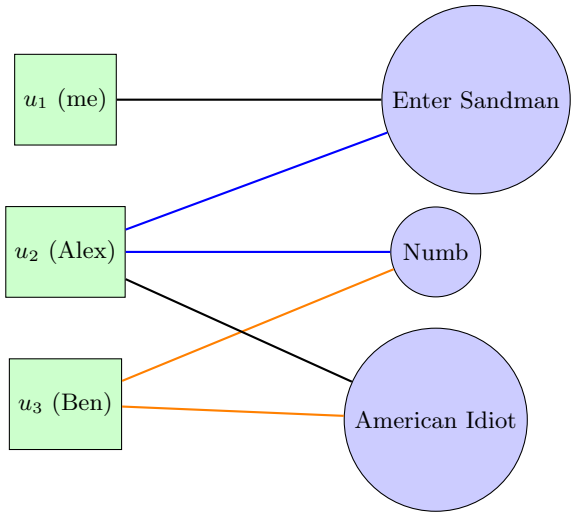

Figure 2: Bipartite graph: I love Metallica, Alex loves Metallica+Linkin Park, Ben loves Linkin Park+Green Day.

3 Adjacency matrices reveal hidden connections

Turn my song graph into matrix M (Enter Sandman=A, Numb=B, American Idiot=C):

$$M = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Square it to count 2-step paths:

$$M^2 = M \cdot M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$(M^2)_{AC} = 1$ means ONE path: Enter Sandman → Numb → American Idiot. M^3 would count my full path to Blink-182!

Normalised version $D^{-1/2}MD^{-1/2}$ (D=degree matrix) finds long-range similarity through diffusion.

4 User similarity - finding my musical twins

Connect users who share songs. Jaccard similarity:

$$\text{Sim}(u_i, u_j) = \frac{|P_i \cap P_j|}{|P_i \cup P_j|}$$

Me (u_1): 20 Metallica + 10 others = 30 songs. Alex (u_2): 15 Metallica + 15 Linkin Park = 30 songs. Shared: 15 Metallica.

$$\text{Sim}(\text{me}, \text{Alex}) = \frac{15}{35} \approx 0.43$$

Ben shares 10 Linkin Park + 5 Metallica with Alex = Sim(me,Ben)=0.3. Similar users vote for my recommendations!

5 Collaborative filtering - score every song

Score for American Idiot (s_k):

$$\text{Score}(s_k) = \sum_{j:u_j \text{ likes } s_k} \text{Sim}(\text{me}, u_j)$$

Alex (0.43) + Ben (0.3) love American Idiot: Score=0.73. All the Small Things gets contributions from Green Day fans similar to Ben. Highest scores → Discover Weekly!

Item-based filtering: similarity between Enter Sandman and Numb directly.

6 Big matrices - automatic taste learning

Imagine a massive Excel sheet: 600 million rows (users), 100 million columns (songs). Me-Enter Sandman = 1 (listen), me-Blink = 0 (haven't heard).

Too huge! So they "factorize" it: $A \approx U \times S$

U = my taste in 100 numbers. S = each song in 100 numbers. Predicted liking = dot product:

$$\text{my score for Blink} = U_{\text{me}} \cdot S_{\text{Blink}}$$

Computer learns these 100 numbers automatically from everyone else's listening. My Metallica taste naturally matches Blink's pop-punk energy!

7 Probability - Will I actually play it?

Turn scores into click probability:

$$P(\text{play}) = \frac{1}{1 + e^{-\text{score}}}$$

Score 2 = 88% chance I'll play. Score 0 = 50/50. Score -2 = 12% chance. Perfect for ranking my playlist!

8 Trees - smart playlist ordering

Graphs find songs. Matrices score them. Trees decide playing order.

Picture a flowchart after Enter Sandman:

- "High energy song?" → YES
- "Evening listening?" → YES
- "Ready for punk?" → YES → Numb
- "User skips rarely?" → YES → American Idiot

Maths guarantee: tree with 10 questions has exactly 9 branches, always finishes, no loops forever.

9 The fancy Spotify sound features

Spotify scans every song for "DNA":

- Energy: Metallica=92%, Blink=78%, Green Day=82%
- Danceability: Linkin Park=73%, Blink=68%
- Happiness: Metallica=32%, Blink=72%

Songs with similar DNA get connected. Enter Sandman's high energy → Numb's medium energy → Green Day's punk energy → Blink.

Computer crunches these 12 numbers per song to find patterns automatically.

9.1 How it all fits together

1. **Graphs** find candidates (Metallica→Linkin→Green Day→Blink)
2. **Similar users** vote on scores
3. **Matrices** learn my hidden taste automatically
4. **Probability** ranks click chances
5. **Trees** order my perfect playlist
6. **Sound DNA** provides real musical connections

10 Conclusion

Next time All the Small Things blasts after Enter Sandman, I hear the full system. Graphs draw paths across genres. Similar fans vote. Matrices discover taste I didn't know I had. Trees sequence perfectly.

Same system powers YouTube, Netflix, Instagram. A 16-year-old gets it, 600 million people live it daily.

Am I discovering music, or just clicking predictions? Either way, Spotify seems to understand music taste, continues to entertain us, and the maths in my AirPods works insanely well.