

CollabGraph – Spotify Collaboration Network Project Overview

What the project is

CollabGraph is a Python-based data analytics project that generates an artist collaboration network using the Spotify Web API. Given a seed artist ID, the program builds a two-hop network graph where nodes represent artists and edges represent collaborations based on co-appearances on the same track. An edge between Artist A and Artist B exists if both appear in a track's artist list. Edge weight equals the number of shared tracks.

Project goal (the product goal)

The goal is to produce shareable, data-rich visualizations of collaboration networks starting from any seed artist. **Inputs**

- Seed artist Spotify ID (required)
- Optional scope parameters to control graph size:
 - Maximum number of hop-1 collaborators retained
 - Optional minimum release year filter
 - Optional inclusion of "appears_on" albums

Outputs

- 1) Interactive network graph (HTML) with zoom, pan, and hover tooltips
- 2) High-resolution static image (PNG) suitable for social media posting
- 3) CSV datasets used to build the graph:
 - nodes.csv – artist attributes (name, popularity, followers, hop level, graph metrics)
 - edges.csv – collaboration links and weights
 - edge_tracks.csv – track-level evidence for each connection

Your personal goal

This project is designed as a Python re-entry and skills-refresh exercise. It focuses on: - Writing clean, readable Python code you understand

- Working with APIs and pagination
- Structuring data pipelines and datasets
- Building visual analytics outputs
- Practicing reproducible project setup suitable for GitHub

High-level execution plan

- 1) Set up a clean repository and Python virtual environment
- 2) Authenticate with Spotify API using spotipy and environment variables
- 3) Fetch seed artist metadata
- 4) Retrieve the seed artist's albums and tracks
- 5) Extract collaboration data from track artist lists
- 6) Build hop-1 collaborator set and apply scope limits
- 7) Crawl hop-1 artist catalogs to build hop-2 edges within allowed set
- 8) Construct nodes, edges, and edge_tracks datasets
- 9) Compute basic network metrics (degree, centrality, communities)
- 10) Generate interactive HTML network graph
- 11) Generate high-resolution static PNG graph
- 12) Package outputs into a timestamped results folder
- 13) Optionally publish interactive graph using GitHub Pages

End result

A fully reproducible, visually compelling Spotify collaboration network that combines data engineering,

analytics, and visualization — while reinforcing practical Python skills in a real-world project.