Prof. Rachlin needs new music!

DS 4300: Large-Scale Storage and Retrieval (Prof. Rachlin)

Problem Description:

A band I really like is *The Strokes,* particularly their album: *Is This It.* In fact, I'm listening to this album as I type this! But I need to expand my musical tastes and I'd like you to help me by using a graph database (Neo4J) to build a song recommendation system.

The Data:

I downloaded Spotify data from Kaggle containing information on about 114,000 songs and their musical properties. The dataset includes fifty-two songs by the Strokes including all of the songs on the *Is This It* album.

Your Task:

Model some or all the Spotify data as a graph database. Nodes are songs and their properties. Edges connect *similar* songs, where similarity is defined however you like. Then, using a cypher query, or a series of cypher queries, generate FIVE song recommendations for Professor Rachlin. Do NOT recommend any songs by *The Strokes.*

Some Hints and Guidelines.

This is a very large dataset, so connecting every pair of songs via an edge with a similarity score property is probably not feasible. Your graph database would have $(114{,}000)^2 = 13$ billion edges. Instead, you are free to sample the dataset any way you like but be sure that your sample includes some or all of the songs from the *Is This It* album. Connect any two songs via an edge if they meet some similarity metric threshold. For example, you might sample 1000 songs, and pre-compute a Euclidean distance metric between every pair of songs, including an edge between two songs if those two songs are sufficiently similar.

IMPORTANT CLARIFICATION: I am asking you to build a general-purpose song recommendation system that could recommend songs based on liking other songs in the database. You will then *test* your approach by recommending songs for me based on the fact that I like The Strokes. Do not simply build a song recommendation capable of *only* recommending songs for me. For example, building a limited network of similar songs connected only to the songs in the *Is This It* album would not be very useful! By building a larger network it also opens the possibility of exploring the differences in genre-specific sub-networks. Are the patterns of interconnection between Hip-Hop songs different from those of Pop? If we built such a network to include

release dates, could the resulting graph give us insights into the history of popular music or musical trends and influences?

What to Submit

1) Your code including pre-processing code (implemented in a programming language of your choice) and all cypher queries that you executed to build your graph network and generate the recommendations.

2) Code documentation sufficient to reproduce your results, including:

    a) How you sampled the data

    b) The full graph data model: Specifically, what are your node and relationship properties and labels?

    c) Explanation of your recommendation algorithm

3) Poster Slide: ONE slide (in PDF format, not PPTX!) that includes:

    a) A graph visualization connecting songs by the Strokes to songs to your recommendations. In other words, a graph representation showing the main results of your queries.

    b) A 1-paragram summary description of your recommendation approach.

    c) Identifies the TOTAL SIZE of your song graph model: # nodes, # edges

    d) A list of the FIVE song recommendations (Artist, Album, and Title).

    e) Your song similarity metric. (What is your rule for connecting two songs?)

    f) We'll share our posters and listen to (what I hope will be) a lot of good music in an upcoming class.