Spotify Artist Recommendation Engine

Team 039: Ziyuan Cao, Pallavi Eranezhath, Tejas Lokeshrao, Harita Patel, Aditya Sasanur

INTRODUCTION

- **Goal:** Create music recommendation system
- **Monotonous Recommendations:** Users encounter repetitive music recommendations on streaming platforms
- **Privacy Constraints:** Privacy limitations hinder the ability of recommendation systems to adapt to dynamic music preferences

DATA

- **Spotify API:** Collect artist information and related artist information
- **# of Records:** 20^n, where n is depth of artist network; with a depth of 5, **3.2 Million records extracted**
- **Wikipedia API:** Gather key information related to each artist from their Wikipedia page
- **User Surveys, Spotify Wrapped Statistics**

EXPERIMENTS

Methods

- Utilized Spotify Wrapped data to categorize users into 3 main artist listening ranges: 0-250, 250-1000, 1000+
- Formed focus groups with 10 participants per range for comprehensive evaluation
- Collected insights on ease of use, effectiveness in recommending artists, overall satisfaction, and comparison to Spotify's recommendation tool

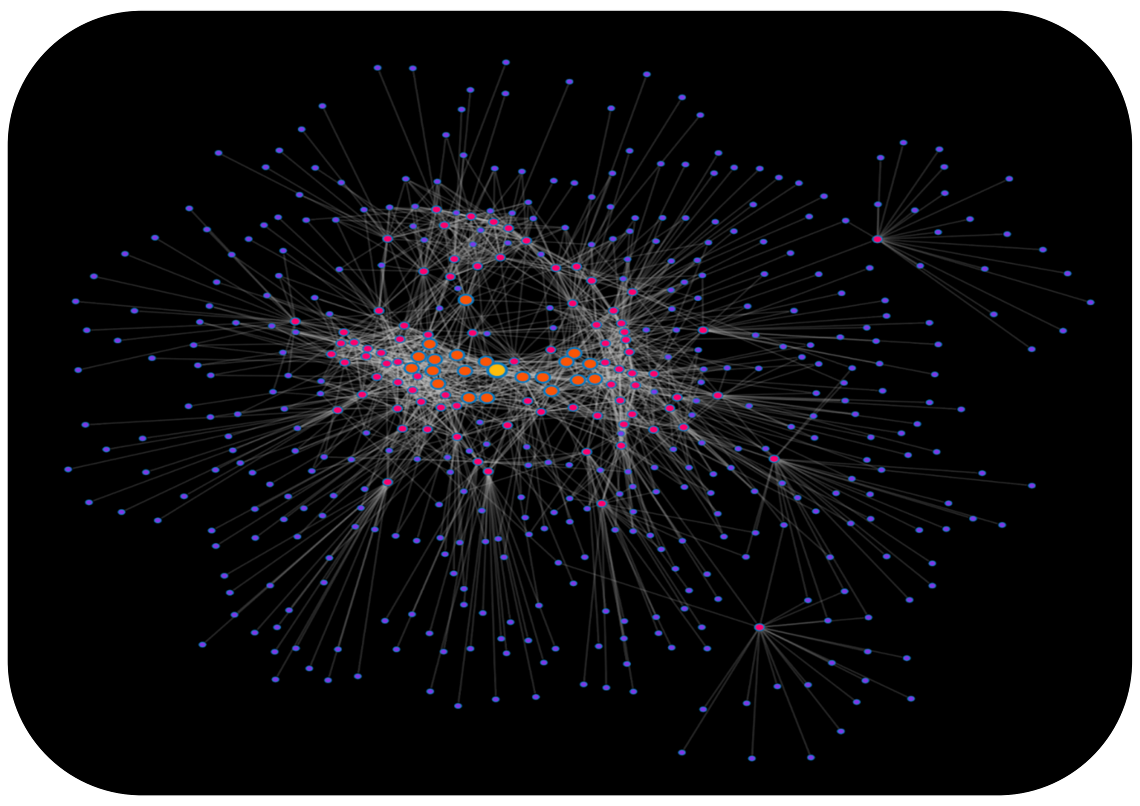
APPROACH

Methods

- Hybrid approach using collaborative filtering and content-based methods
- Introduction of active learning through natural language queries for user engagement
- Dynamic modification of the user-music interaction matrix based on user responses
- Content-based model scans music database for tailored suggestions, ensuring diversity
- Integration of active learning into a hybrid system for a personalized and interactive music discovery experience.

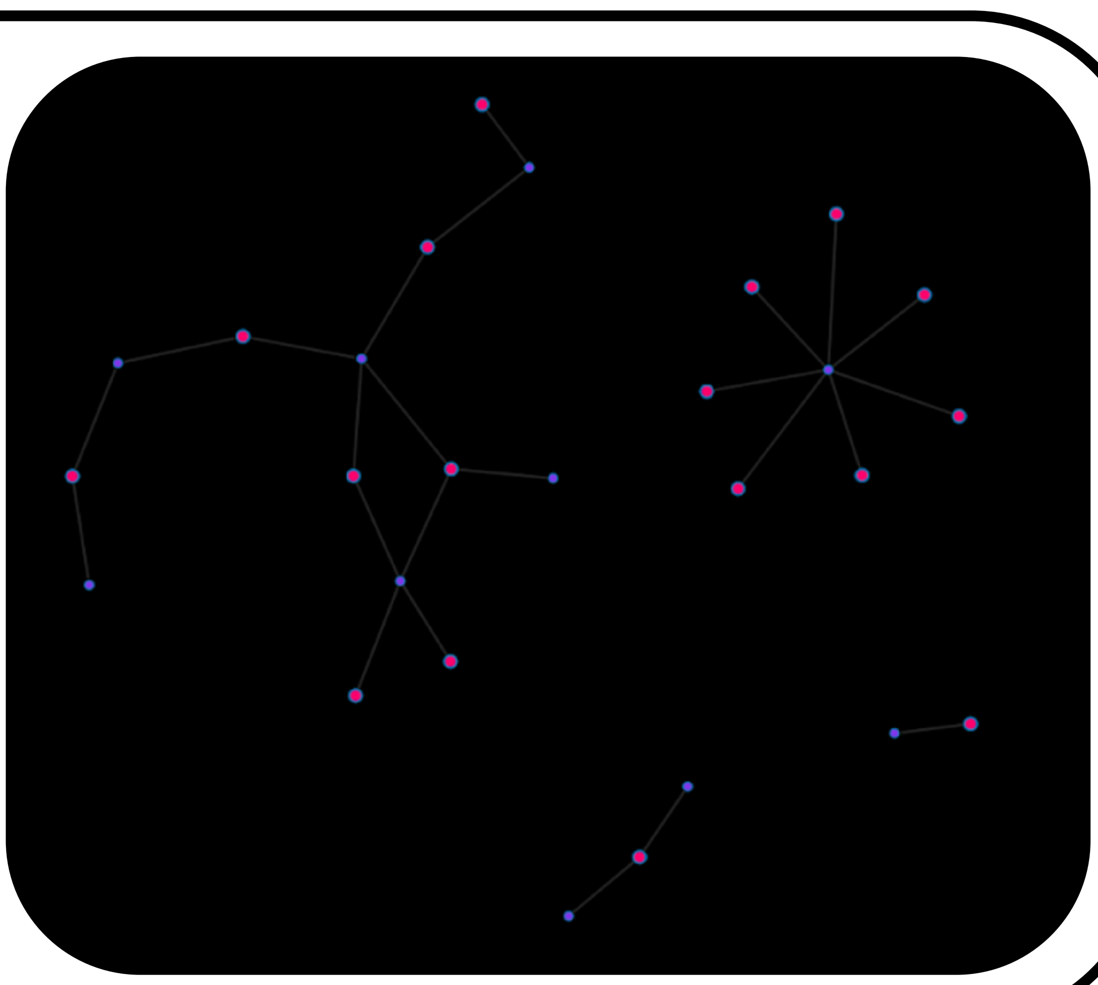
Artists Network Visualization Tool

- Allows users to discover similar artists to their favorites
- User inputs artist name and depth level, initialing a BFS algorithm using the Spotify API to discover related artists to initial artist
- Edges are created between connected artists and finally assembled into a JSON Graph object
- Created aesthetic and interactive user interface with Alchemy.js library into HTML website



Natural Language Querying

- Users can filter to see a sub-network that only includes artists that meet their interest the most
- Achieved with a ranking and filtering approach
- Integrated Wikipedia API to acquire key information about artists and GPT-4 to extract key highlights
- Convert user query into vector representation with Sentence-BERT
- Ranked top 10 artists by cosine similarity



RESULTS

Average Survey Responses for Artist Network Tool	Users with 0-250 Streamed Artists Average Rating	Users with 250-1000 Streamed Artists Average Rating	Users with 1000+ Streamed Artists Average Rating
Ease of Use	9.6	9.3	9.6
Effectiveness	9.6	8.3	8.1
Satisfaction	9.4	8.5	8.2
Comparison to Spotify	0.3	1.2	1.6

Average Survey Responses for Natural Language Query Network Tool	Users with 0-250 Streamed Artists Average Rating	Users with 250-1000 Streamed Artists Average Rating	Users with 1000+ Streamed Artists Average Rating
Ease of Use	9.2	9.6	9.6
Relevance of Recommendations	8.4	9.6	9.8
Satisfaction	8.2	9.8	9.8
Comparison to Spotify	3.2	3.5	3.7

- Artist Network Tool
  - Users with fewer artists found the tool more effective
- Complexity impacted satisfaction, with users familiar with more artists giving lower satisfaction scores
  - Rated slightly better than Spotify recommendations
- Natural Language Query Tool
  - Users with more artists found natural language query Tool more effective
  - Relevance of recommendations was perceived higher by users with broader music tastes
  - Well-favored over Spotify in recommending artists across all user groups
- Simplicity: both tools received better ratings when they were less complex and users did not have to parse through extensive networks
- Average ratings differed by at most 0.2, indicating a direct correlation between usefulness & satisfaction
- As degree connections increased, artist network became cluttered, leading to challenges in usability
- Future work: branch pruning to allow users to remove unwanted connections