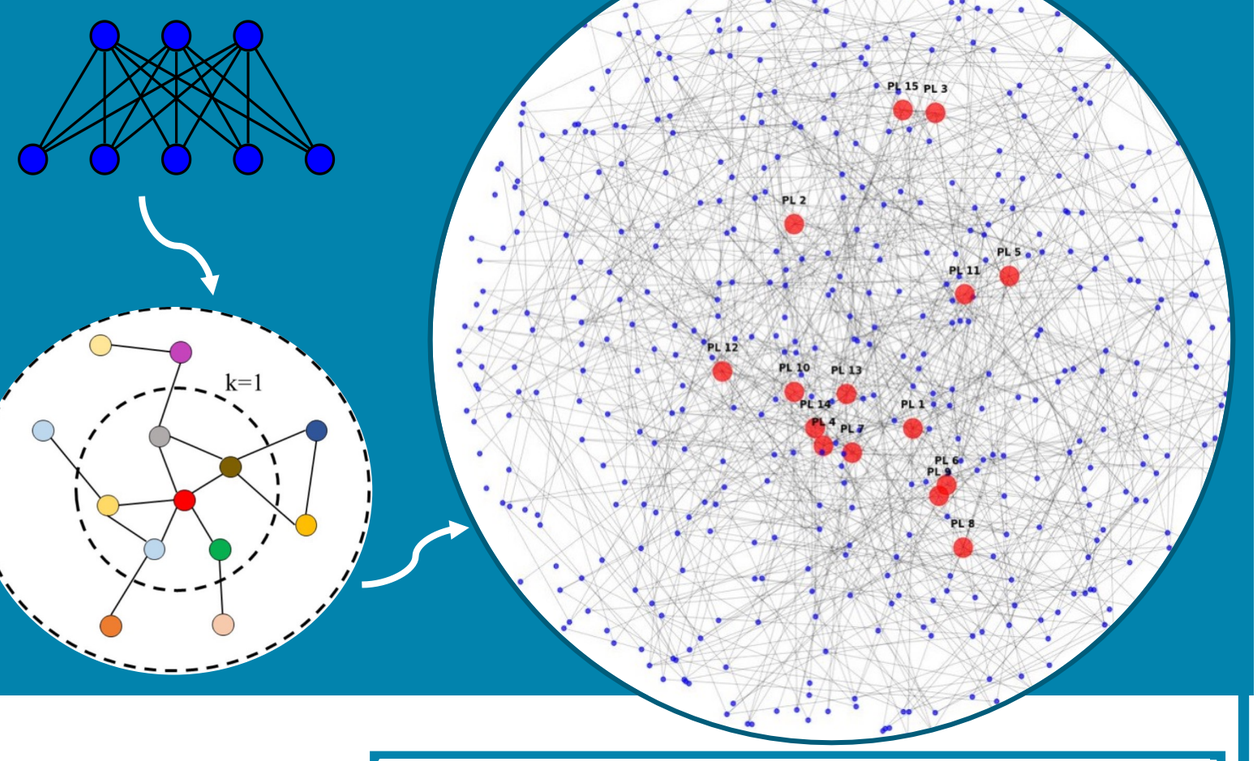
G-PLAY: Graph-based Playlist Continuation using Neural Collaborative Filtering

Abhishek Deshpande
Ayush Sharma
Bhavya Sankhla
Rituraj Ojha

Introduction

Music recommendation systems have become an integral part of our daily lives, and collaborative filtering has been a popular technique used to predict user preferences. However, collaborative filtering suffers from several bottlenecks, including the lack of useful metadata, sparsity, and scalability issues. Graph-based collaborative filtering methods have gained popularity due to their ability to handle these bottlenecks and model complex relationships. In this project, we aim to use LightGCN, a state-of-the-art graph-based collaborative filtering method, to continue Spotify playlists by predicting which songs a user is likely to add to a playlist based on their existing preferences and playlist history. LightGCN leverages the graph structure to handle sparsity and scalability issues and incorporates neighbourhood information into nodes using an aggregation strategy. Additionally, it is capable of handling cold-start scenarios by generating embeddings for new users or items with no interactions by leveraging the embeddings of existing nodes in the graph. By improving the accuracy and scalability of music recommendation systems, we aim to provide a more personalized and satisfying music discovery experience for Spotify users, and ultimately contribute to the development of more effective and efficient recommendation systems in other domains.



Data Science Pipeline

Spotify Million Playlist Dataset

Data Extraction, Cleaning & Analysis

- Spotify Million Playlist dataset
- 40 GB of JSON files
- Each file contains 1,000 playlists
- Each playlist contains varying number of tracks

graph-tool — Graph Generation

- Generate graph from JSON files
- Track node attributes
- Retrieve Largest Component
- Find central nodes and generate n-hop neighborhood

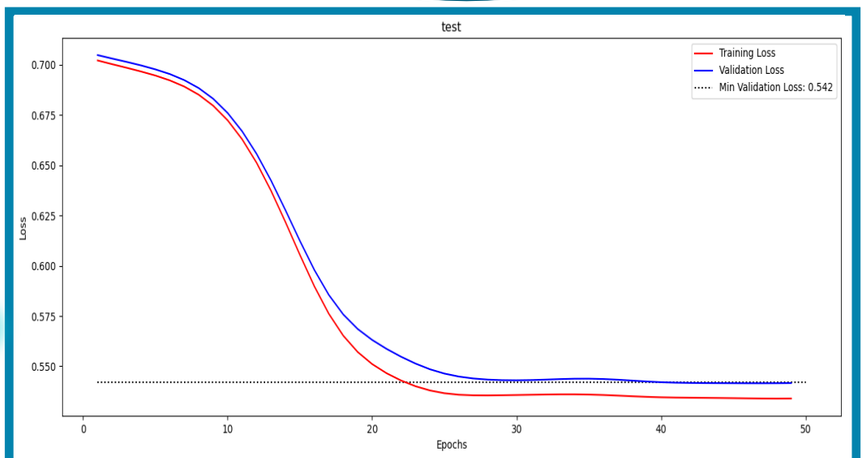
Preprocessing — NetworkX Network Analysis in Python, DeepSNAP

- Dataset Generation for Link Predictions
- Supervision and Message Passing edges
- Sample positive and negative edges for train, validation & prediction

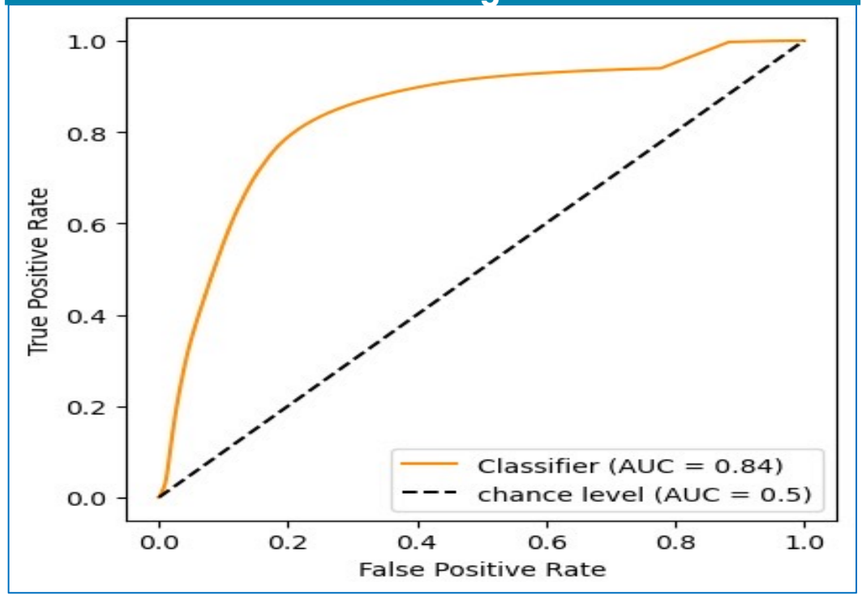
PyG, PyTorch geometric — Model Training and Testing

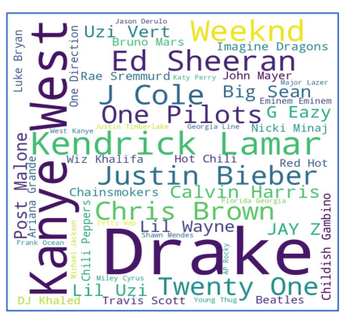
- LightGCN Architecture with modified loss function and aggregation method for generation of node embeddings
- Model training, validation and evaluation

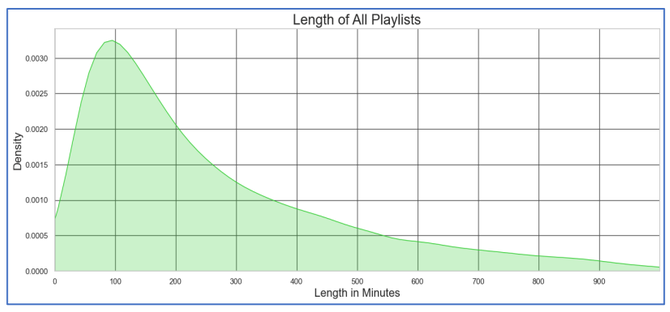
- Inference on playlist nodes to extend connections to unconnected, potential track nodes.
- Interfacing results with frontend interface



Train/Validation Loss Curve Recommending Spotify Playlist Tracks with Neural Collaborative Filtering Using Graph Machine Learning



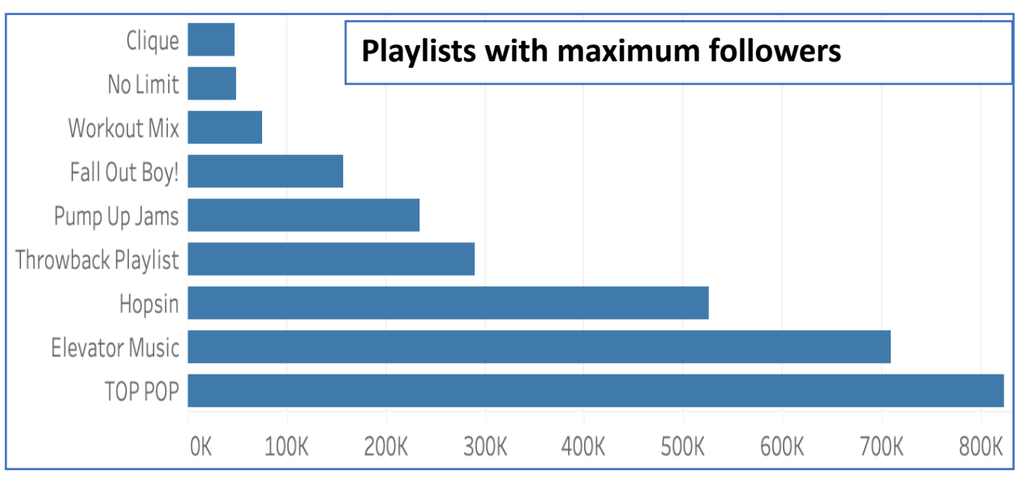


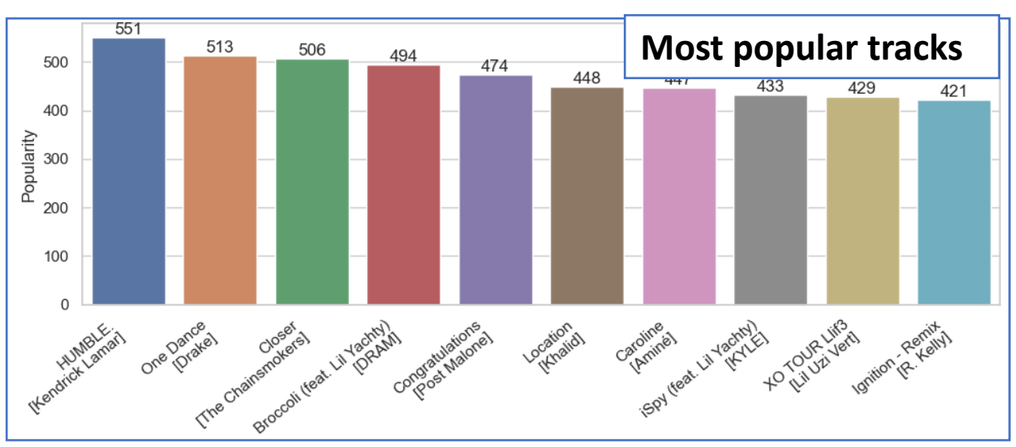


Exploratory Data Analysis

OBSERVATIONS

1. There are 12000 unique playlists with total of 39315 artists and 193090 unique tracks.
2. "HUMBLE" by Kendrick Lamar is the most added song in Playlists, followed by "One Dance", "Closer, "Broccoli (feat. Lil Yachty)", and "Congratulations".
3. Artist with the largest number of songs added to playlists is Drake.
4. The most followed playlist is "TOP POP" having 15842 followers.
5. The playlists have an average of 261 minutes of listening time.





Achievements

- Trained a *LightGCN* Pytorch Geometric (PyG) model using Binary Cross-Entropy (BCE) loss.
- Achieved ROC-AUC score 0.84 for testing.
- Able to converge within 50 epochs due to our optimized approach to model training.
- Developed a playlist recommendation webapp using Python, Flask framework, JS, HTML & CSS.

Recommendation WebApp