Song Recommendation

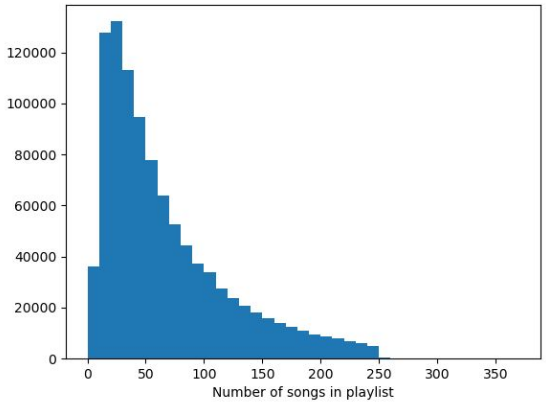
•••

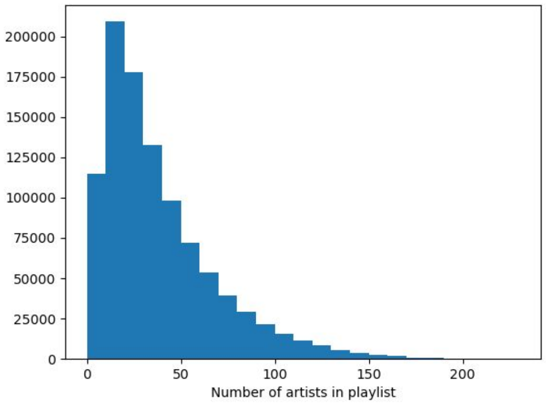
Playlist continuation based on Spotify data

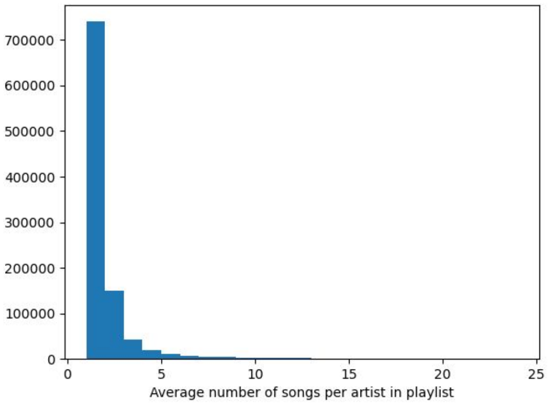
Bhasitha Dharmasena
Arlyn Reese Madsen
Ernesto Martinez
Masahiro Nakahara
Greg Taylor

Data Gathering

- “Million Playlist Dataset”
  - Playlist attributes: name, description, number of tracks, number of followers, etc.
  - ~ 2.2 million unique songs, and ~ 290 thousand unique artists
- Spotify API
  - Python library “spotipy”
  - Audio features of each song (“energy”, “danceability”, “liveness”, etc.)
  - Genres for each artist

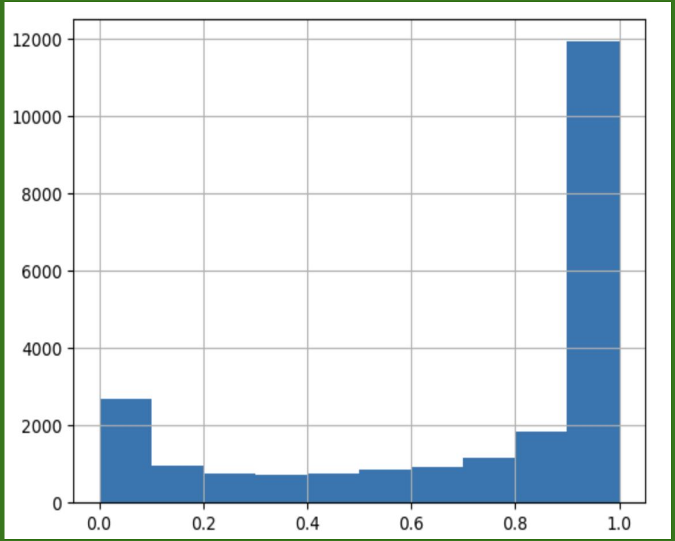






Preprocessing and Vectorization (NLP)

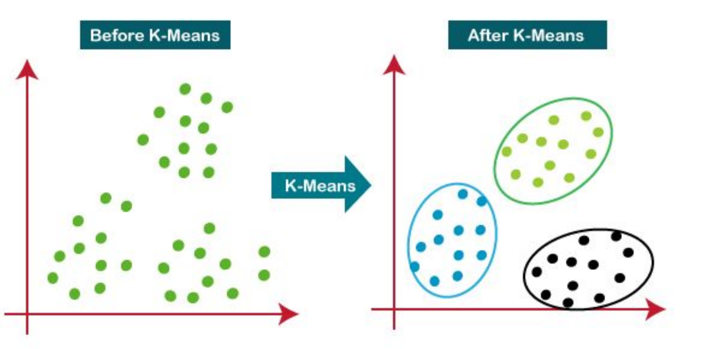
- Scraped lyrics from Genius.com using the LyricsGenius API.
- Performed sentiment analysis using NTLK's Naive Bayes Analyzer.
- Genres for each artist as provided by spotify
- Vectorized via TF-IDF (Term Frequency - Inverse Document Frequency)



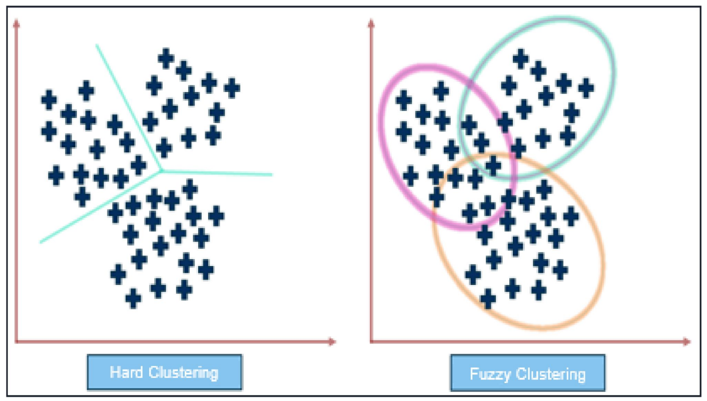
Modeling Approaches:

Unsupervised Learning

- K-means clustering
  - Partition n observations into k clusters
  - Each observation belongs to the cluster with the nearest centroid
  - Generated models for single features and combinations of them to provide flexibility in type of recommendation



- Fuzzy Clustering(Soft Clustering)
  - Data points belong to more than one cluster - gives the relation of each data point to each cluster centers
  - Post Processing to identify most important clusters based on Inputs songs
  - Improved model using Sentiment Analysis - 3 fold accuracy improvement



https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning https://www.mdpi.com/2071-1050/13/11/5823

Model Evaluation

- One way to evaluate our models is to feed it partial playlists and see how well it completes the remaining songs.
- Divide playlist into two parts. The model recommends based on the input songs in the first part and we compute how many of the recommended songs match songs in the second part of the playlist.
- We tested this on every numerical feature of a song obtained from Spotify API. The result was the following order of importance.

Energy > Acousticness > Danceability > Valence > Speechiness > Liveliness = Tempo > Key > Instrumentalness

Web Application

- The packages we used were widgets to provide GUI for Jupyter notebooks and voila to convert a Jupyter notebook into an interactive dashboard.
- User input.
- Then we run the k-means algorithm on danceability and output a list of recommended songs.