

Synesthete: A deep learning engine that sees sound

Vivas Kaul
Sprint 2 Deliverable
8/21/2023

Table of Contents

- Project Overview
- Data Overview and Preprocessing
- Findings from EDA
- Baseline Models and Evaluation Metrics
- Next Steps

Project Overview

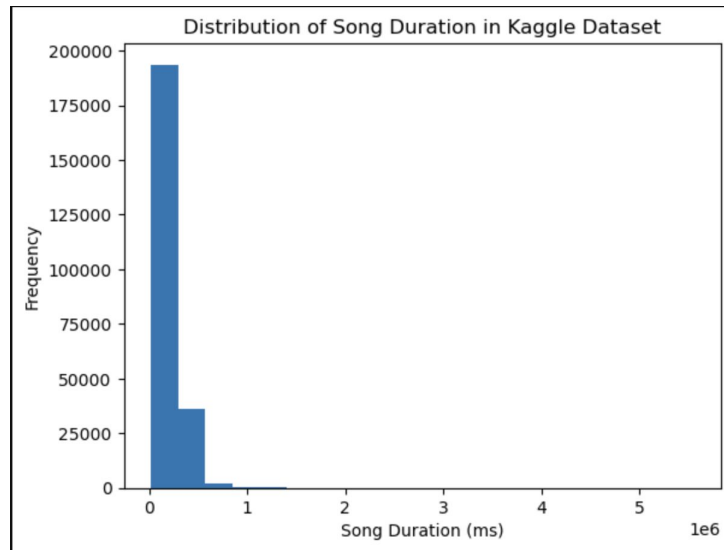
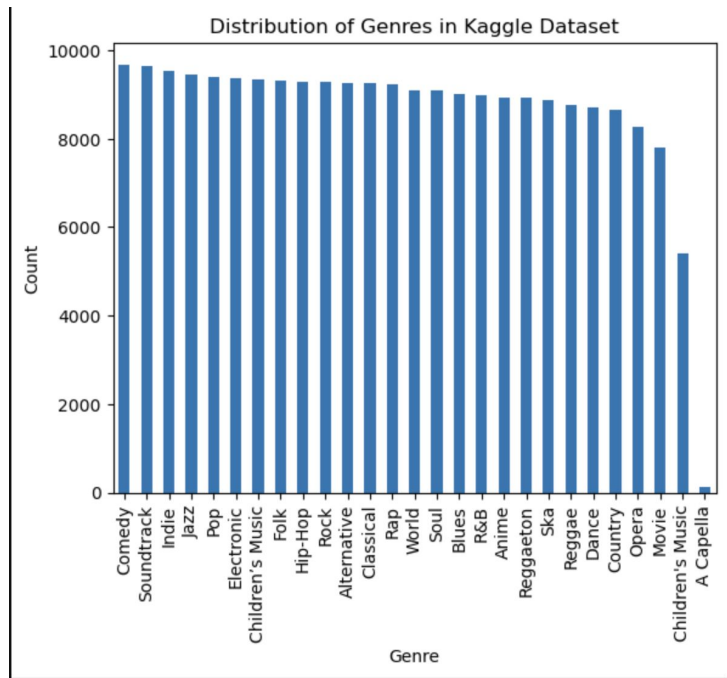
- **Problem Statement:** Recommend similar songs to a user based on a single audio track as an input. The audio can come via a pre-saved audio file or as a recording in real time.
- **Proposed Solution:** Vectorize audio data and calculate cosine similarity. Then present the user with the top 5 matches based on that similarity. A more robust model could also be created by encoding the audio data and letting a CNN determine the similarity.
- **Potential Impact:** Provide the user tools to find similar artists via a standalone app or create value-add for existing app.

Data Overview and Preprocessing

- Kaggle Data
 - 232,725 rows each representing a song
 - No Nulls - very clean
- Vectorized MP3s
 - 11573
- Encoded MP3s using Librosa package
 - Mel Frequency Cepstral Coefficients (MFCC)
 - Mel Spectrogram
 - Chroma Vector
 - Tonnetz Spectrum
- Set `track_id` to category in order to set hierarchy for ordering `track_id` when constructing a playlist

Findings from EDA

Kaggle Dataset



Findings from EDA (cont.)

Kaggle Dataset - Duplication of track_id

```
kaggle_df.track_id.value_counts(ascending=False).head() # some of the track_ids appear more than once  
# This seemed odd considering that the track_id should be a unique identifier.
```

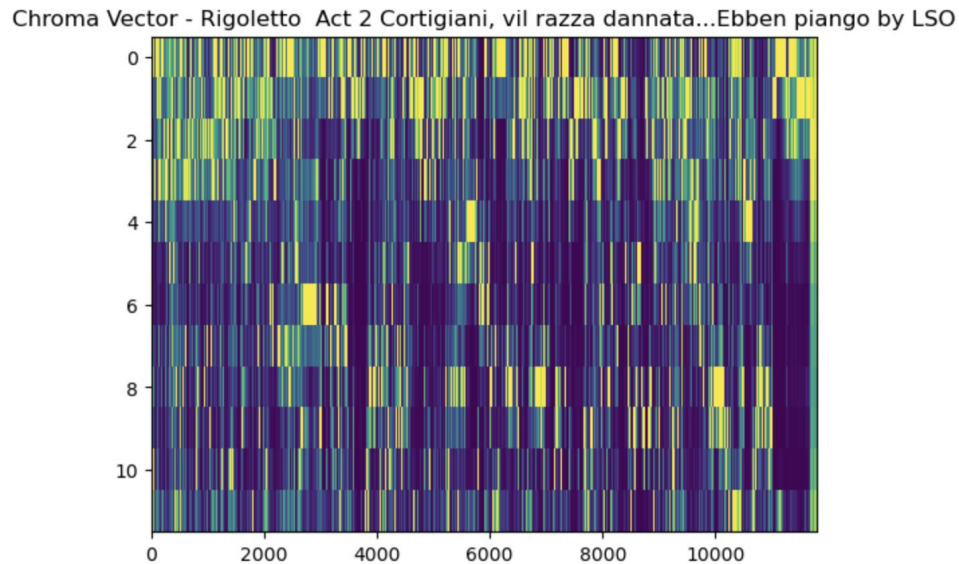
```
3R73Y7X53MIQZWnKloWq5i    8  
0Wy9rA9fJkuESyYm9uzVK5    8  
6sVQNUvcVFTXvIk3ec0ngd    8  
0UE0RhnRaEYsiYgXpyLoZc    8  
6AIte2Iej1QKlaofpCzW1     8  
Name: track_id, dtype: int64
```

```
kaggle_df.loc[kaggle_df.track_id == "3R73Y7X53MIQZWnKloWq5i"]  
# This shows that a single track id does correspond to a single song, but that the song can repeat within the data set  
# based on the genre. All of the other columns are the same though.
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key	livene
5506	Alternative	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	50	0.08	0.65	125133	0.38	0.00	C#	0.
15615	Dance	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.
41367	Folk	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.
55106	R&B	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.
77769	Children's Music	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.
94721	Indie	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.
149585	Pop	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.
225238	Rock	Toro y Moi	Monte Carlo (feat. WET)	3R73Y7X53MIQZWnKloWq5i	61	0.08	0.65	125133	0.38	0.00	C#	0.

Findings from EDA (cont.)

Librosa Vectorized MP3s



Baseline Models and Evaluation Metrics

- Used pairwise from the sklearn.metrics library to calculate cosine similarity for each vectorized mp3.
- Array returned was sorted with each row representing a song followed by it's most similar matches from left to right, by numerical index.
- The indices were then returned to the Kaggle DataFrame to get the original song plus its top five matches
- Recommendations were then evaluated based on summary statistics and listening to them in Spotify

Baseline Models and Evaluation Metrics (cont.)

```
track_recommendations = kaggle_df.copy(deep=True).sort_values(["track_id"]).head(6) # sorts heirarchy
track_recommendations
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness
53452	Blues	The Detroit Cobras	Shout Bamalama	6DQzmTJnaZw4A9lynCGfrM	26	0.03	0.29	121427	0.92	0.36	D#	0.38
32115	Anime	luz	Queen of Heart - Sister Edition	0iPAXRIiRq1vSHbEnjSnZ6	21	0.00	0.52	213070	0.89	0.00	A#	0.32
187364	Ska	Rancid	Something In The World Today	13TbXtmTJ17OuFqW9y7eTw	26	0.01	0.31	153600	0.99	0.00	G	0.25
31113	Anime	FLOW	Howling - Instrumental	0WXb6fBOqbr3GoRKhAz4us	26	0.00	0.49	260333	0.91	0.85	A#	0.09
216175	Country	Rascal Flatts	Why Wait	3uSeT5T2pOEx5fTdYfJasm	44	0.12	0.61	224840	0.77	0.00	D	0.10
15171	Dance	Lizzo	Water Me	4d9GVZP50tSxTfBvHEHUPx	61	0.03	0.67	185595	0.97	0.00	D#	0.18

```
track_recommendations.describe()
# summary statistics for recommendations
```

	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
count	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
mean	34.00	0.03	0.48	193,144.17	0.91	0.20	0.22	-3.46	0.08	132.04	0.61
std	15.43	0.05	0.15	50,358.89	0.08	0.35	0.12	1.32	0.05	30.20	0.23
min	21.00	0.00	0.29	121,427.00	0.77	0.00	0.09	-5.08	0.04	99.99	0.30
25%	26.00	0.00	0.36	161,598.75	0.90	0.00	0.12	-4.53	0.04	114.18	0.53
50%	26.00	0.02	0.50	199,332.50	0.91	0.00	0.22	-3.30	0.06	119.81	0.56
75%	39.50	0.03	0.59	221,897.50	0.95	0.27	0.31	-2.42	0.09	155.84	0.72
max	61.00	0.12	0.67	260,333.00	0.99	0.85	0.38	-1.99	0.18	172.45	0.96

Next Steps

- Determine Evaluation Metric for Model
 - Spotify recommendation engine via API
 - Comparison of CNN with pairwise model
- Training and Construction of CNN
 - Scikit-learn or Tensorflow - most straightforward
 - Pytorch - more robust
- Productionize Model
 - Streamlit App
 - Containerize and mount in Cloud
 - AWS - EC2 and ECR

Questions?