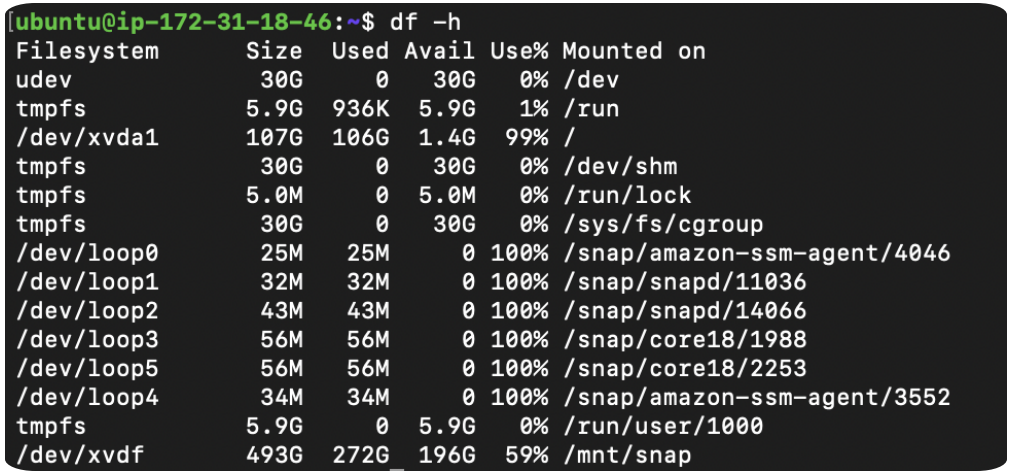
Million Song Dataset

- Ashwani Rajan
- Shubham Thakur
- Sunil Kumar
- Vishwas Prabhu

Dataset description

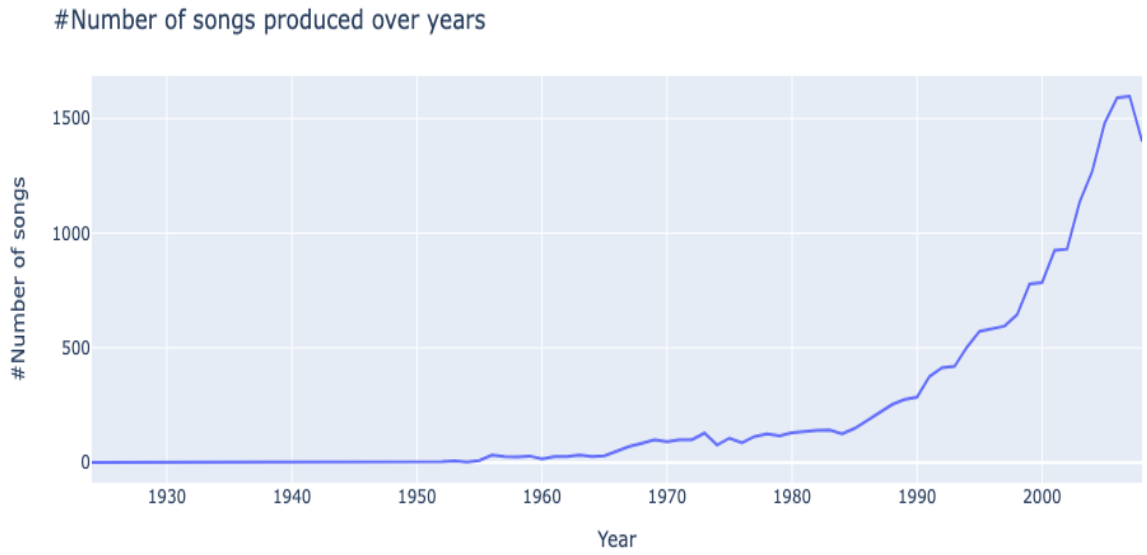
- A Dataset consisting of 1 million songs starting from 1930s
- Overall dataset size: 493 GB

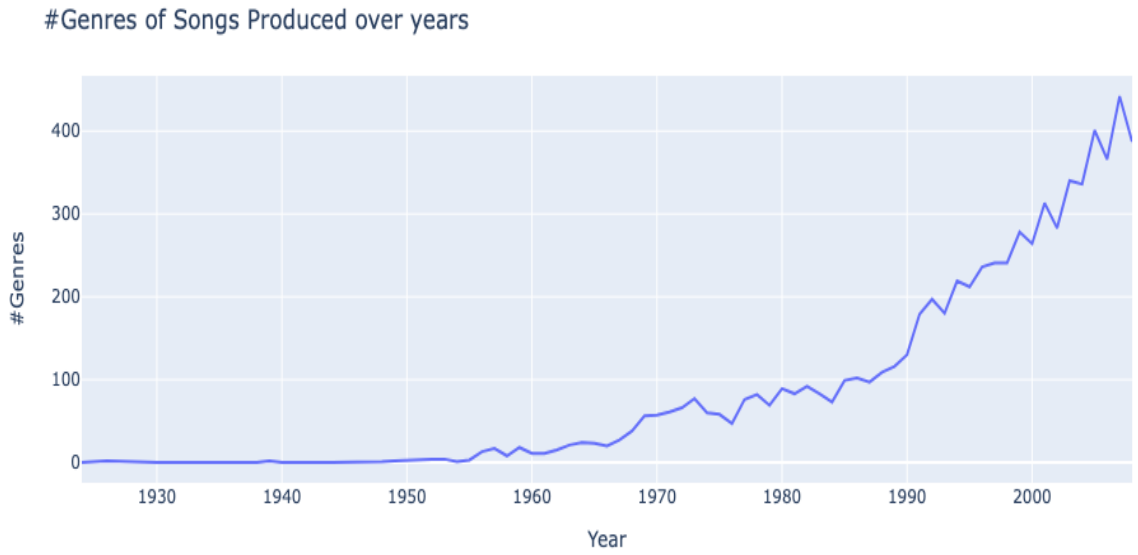
  EBS Snapshot: snap-5178cf30
- Dataset size used (HDF5): 9.2 GB (Estimated)
- Original dataset format: HDF5
- Dataset conversion: Script prepared to parse HDF5 to CSV format. We used c4.xlarge instance.
- Total songs utilized for analysis: 41,182



Number and Genre of songs over time

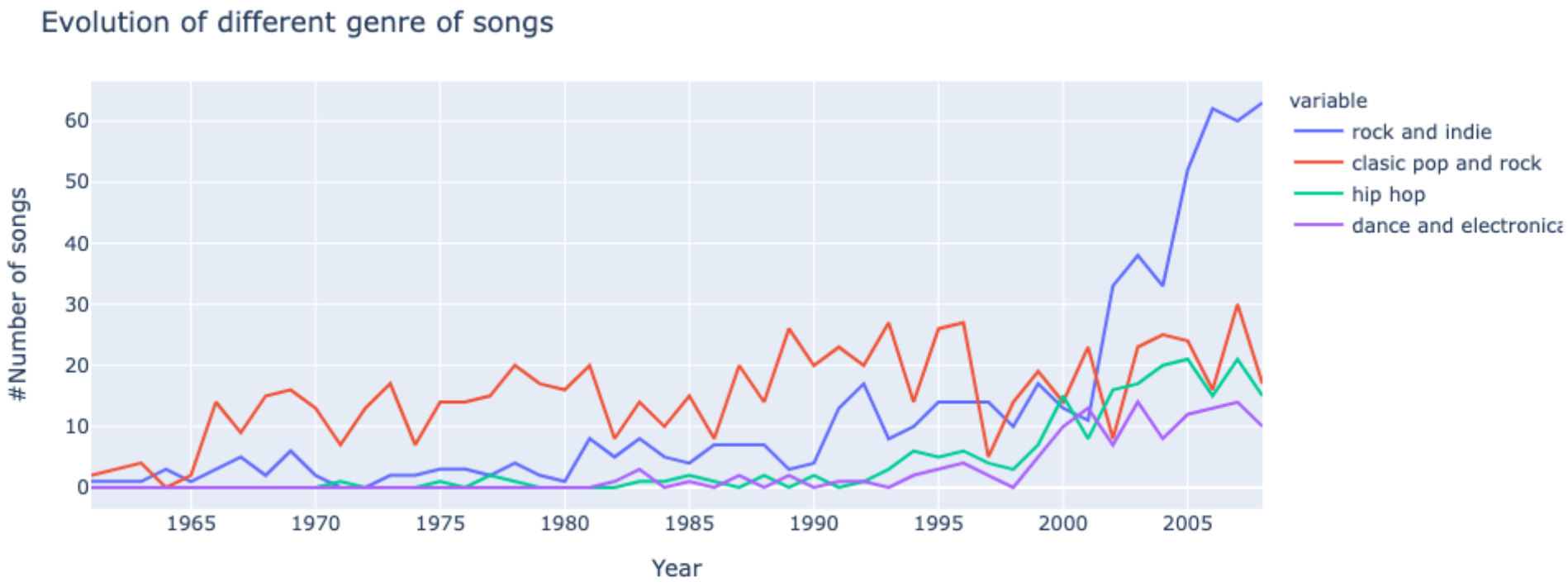
- Number of songs increased exponentially after 1980s
- Diversity in music started increased exponentially after 1970s





Evolution of different genres of songs

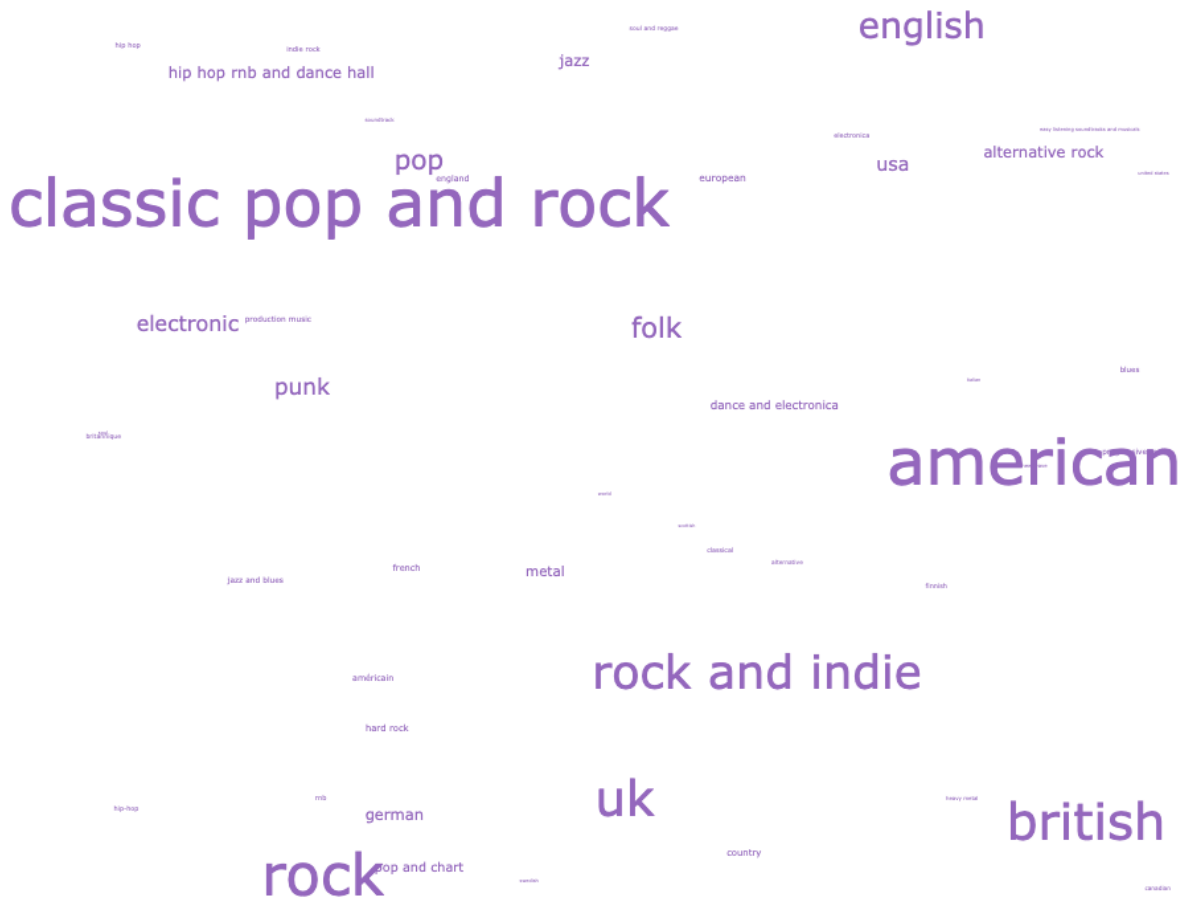
- Rock and Indie music started picking after 2000s
- Classic pop and rock has been consistently popular since 1970s
- Hip –hop and Electronica music started trending in 2000s



MusicBrainz Tags Frequency

Analytics goal: Identifying the most frequent mbtags and their popularity

- *Mb_tags are similar to twitter hashtags which is used to identify group of songs having some entity in common.*

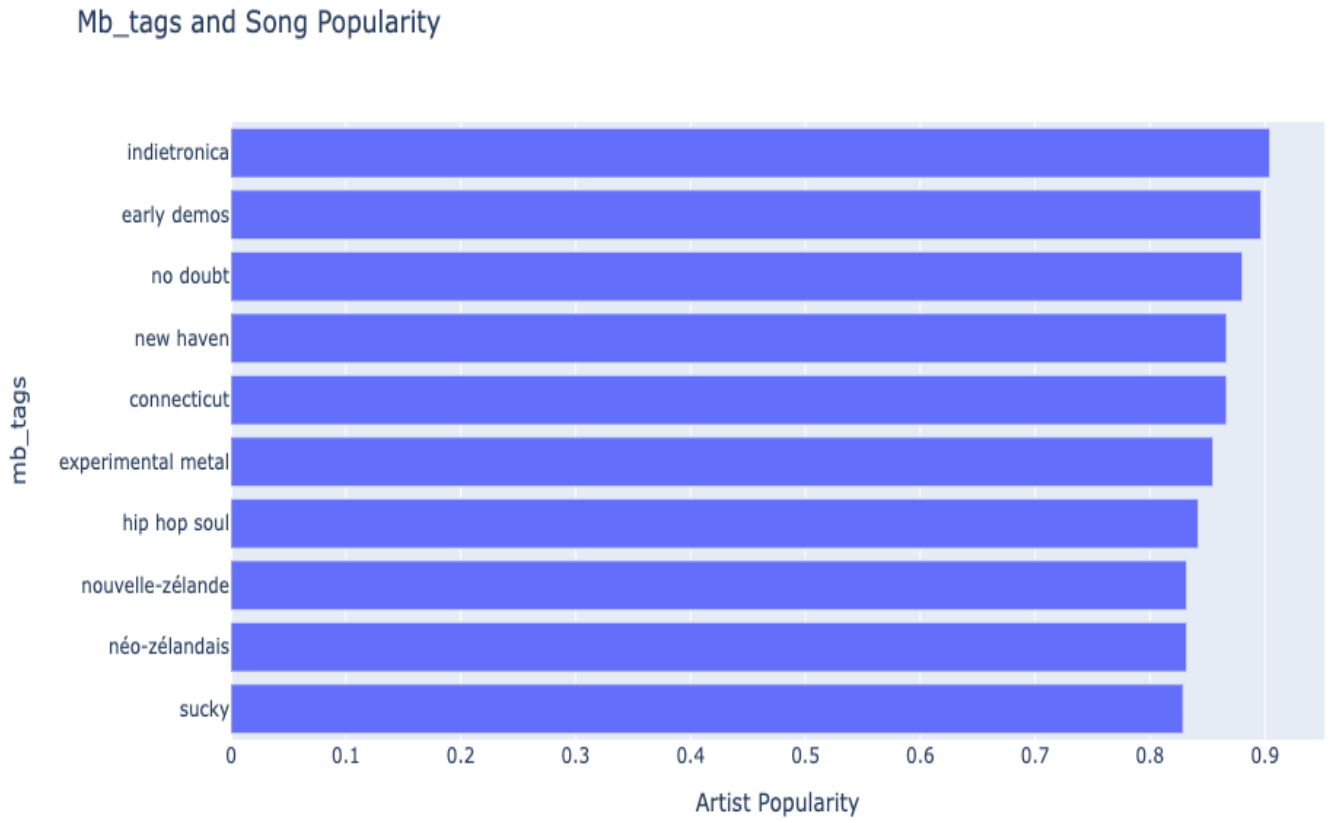


Top tags used in the data

- Classic pop and rock
- American
- Rock and indie
- Hip hop
- Folk
- electonic

Tags with Most Popular Song

Song popularity is based on total number of plays and compared to other songs and how recent those plays are

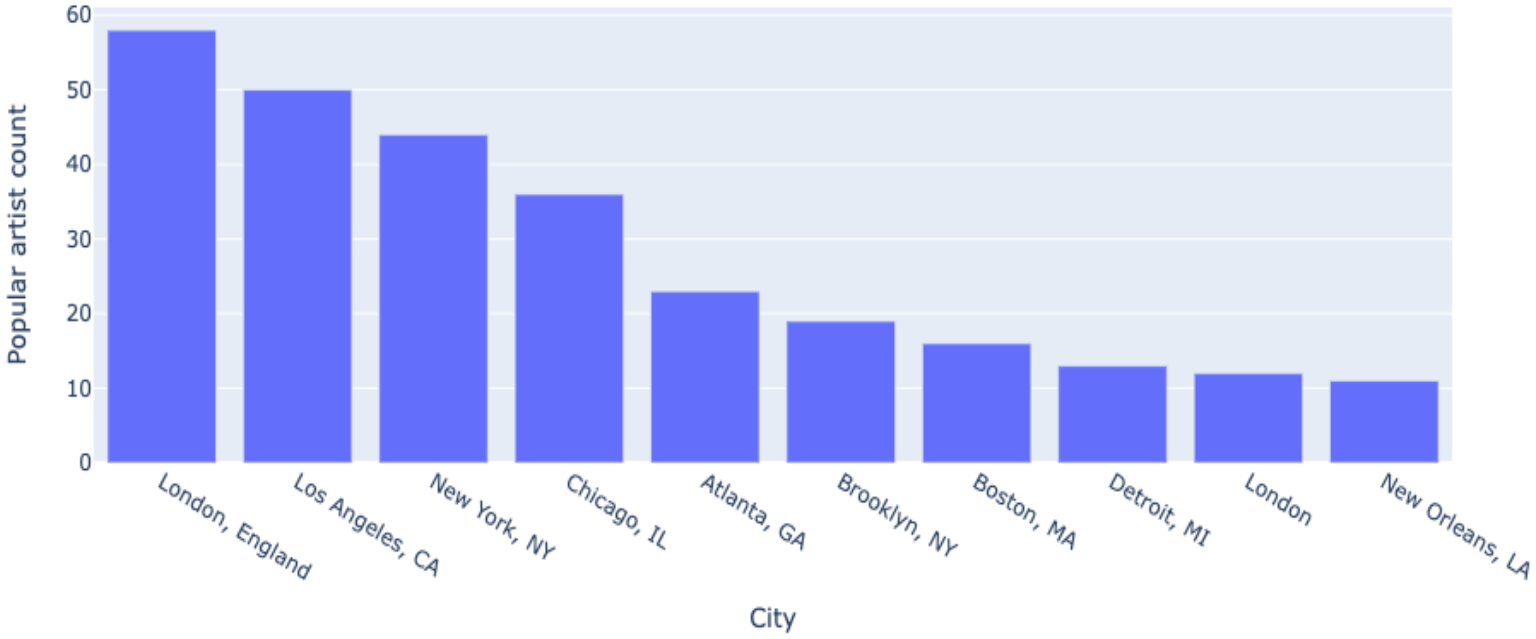


- Indietronica tags have the highest mean Song Popularity
- There is only one tag having average song popularity over 0.9.

* Song popularity is on the scale of 0-1

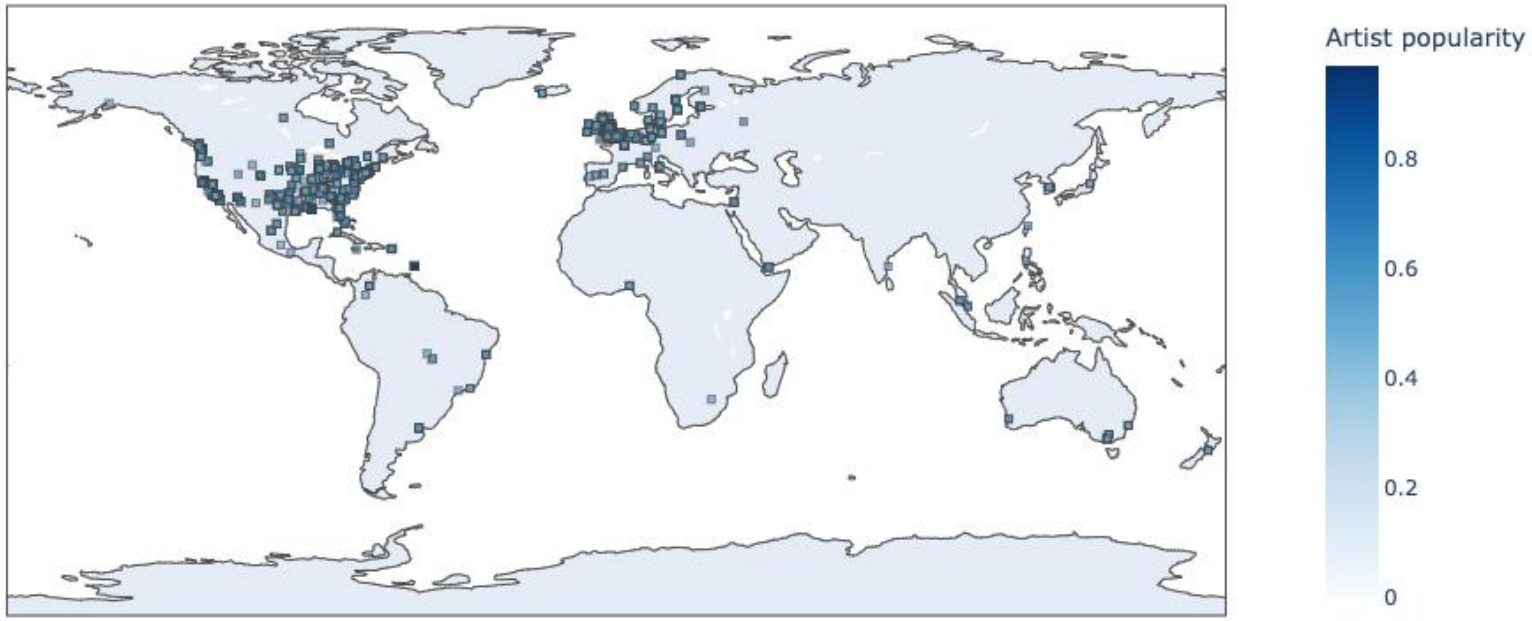
Popular artist count by location (city)

Analytics goal: Analyzing artist popularity based on location and coordinate data.



London is the home to most popular artists as per the dataset

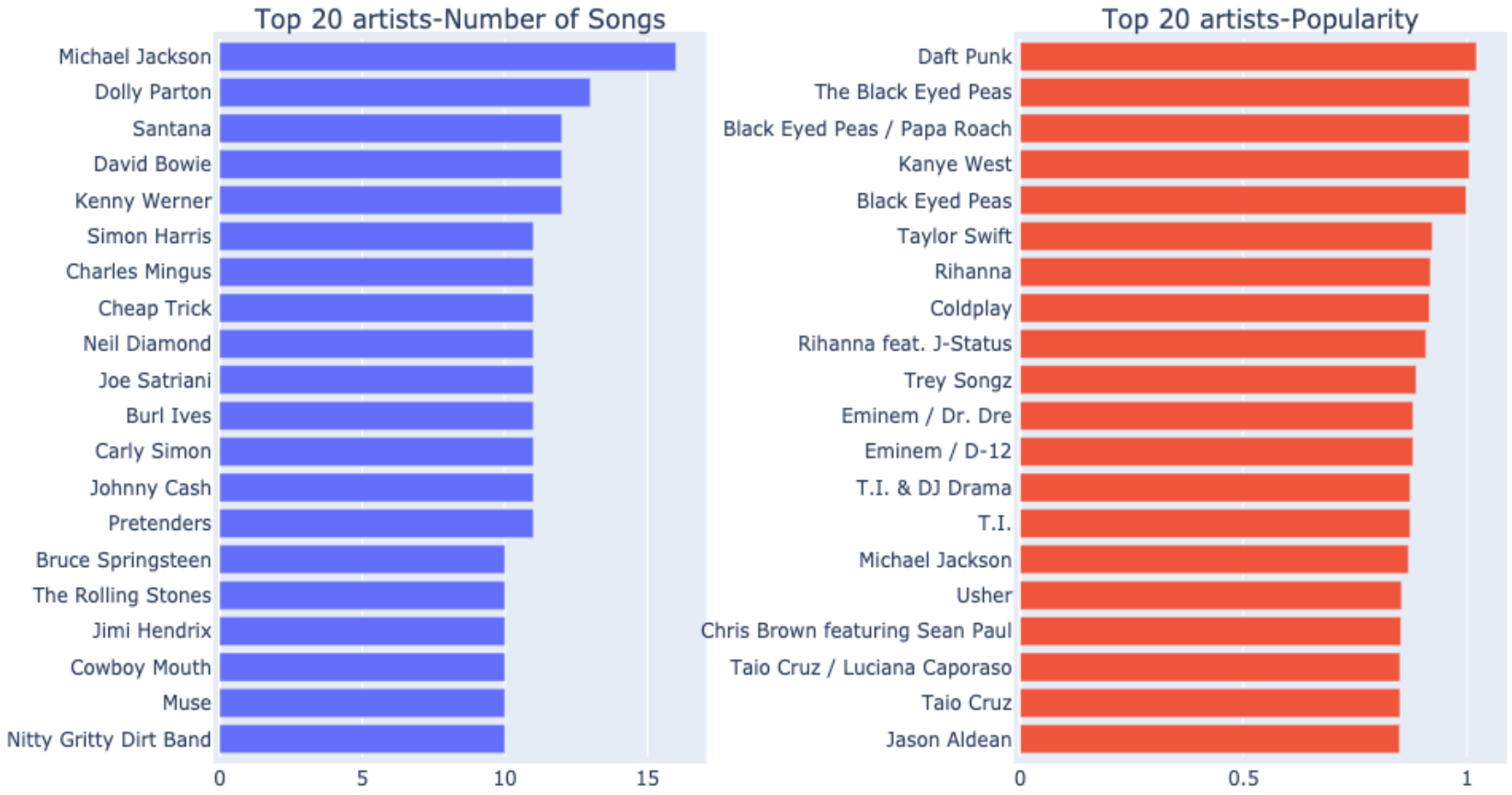
Most number of popular artists in the dataset are from US and European cities



Artist popularity by location

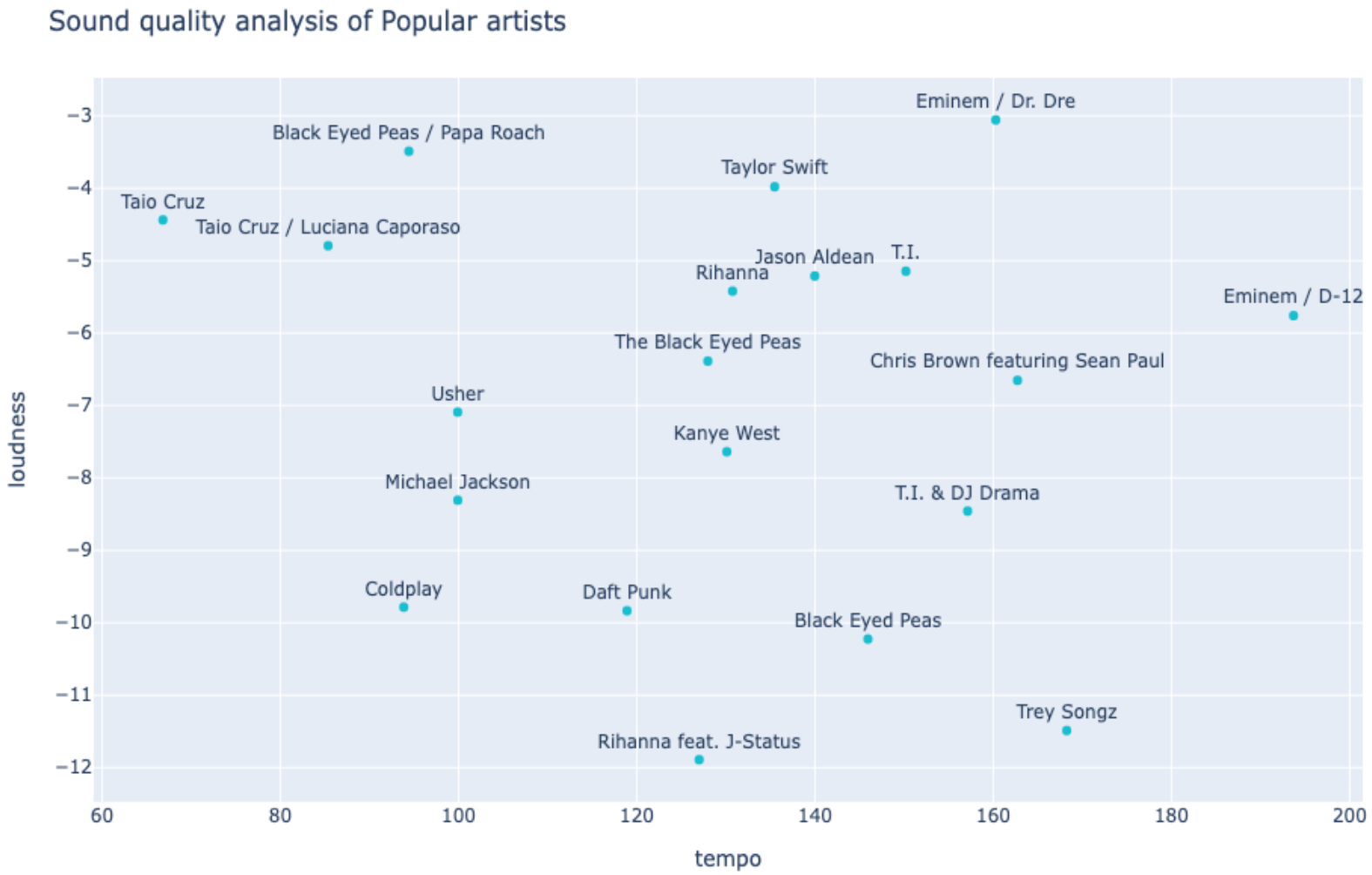
- Most popular artists(Top 1500) plotted by location.
- Most of the popular artists worldwide are from US and European region as per the dataset.

Analytics Goal: Top Artists and their music



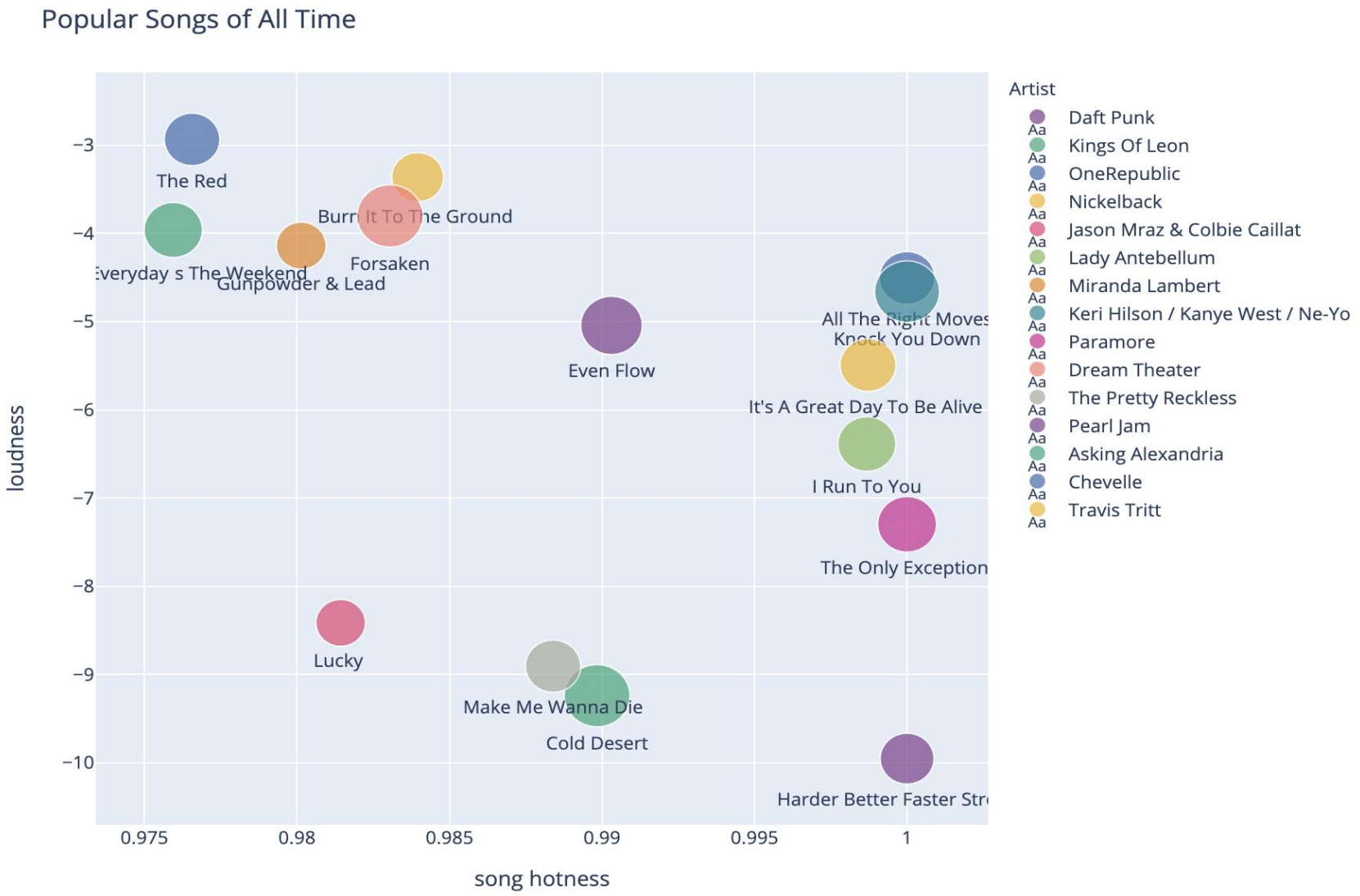
- Artists with more songs are also the more established ones.
- Contemporary/New artists have lesser number of songs but high popularity.

Sound Quality of Popular Artists



- Tempo is the speed of song, (beats per minute)
- Loudness is in LUFS (Loudness Unit Full Scale), which measures the level of perceived loudness.
- For context, The Beatles have a LUFS of –40
- Rappers(Eminem, Sean Paul, etc.) have a higher tempo.

Song Popularity and artist popularity



- Songs with Popularity > 0.97 are taken
- Most popular songs are also perceived as loud.
- Only Daft Punk's "Harder Better Faster Stronger" in most popular songs of all time. Uncorrelated to artist popularity.
- Artist popularity is dynamic and song popularity is permanent
- Bubble Size is song duration

Artist tags and Similar Artists Recommendations

Similar Artists to Eminem:

50 Cent
Ice Cube
Dr. Dre / Knoc-Turn'al
Obie Trice
Cypress Hill
Snoop Dogg
Wu-Tang Clan
Naughty By Nature
D-12
Smut Peddlers

Things done to Improve the code

Change	Reasoning
Usage of reduceByKey instead of GroupByKey().mapValues()	Pairs on the same node/partition with the same key are combined before the data is shuffled
Persistence	reuse an RDD for multiple actions, you can ask Spark to store the content in memory/disk
Num Partitions	Operations involving shuffling data by key across the network

Time Improvement

Name	Execution time (after optimization)	% / improvement in s
Vishwas Prabhu	50 s	46% / 42.04 s
Sunil Kumar J S	44.42 s	13.71% / 7.04 s
Ashwani Rajan	58.127 s	71.4% / 145 s
Shubham Thakur	57.61 s	37% /21.31 s

Cluster Setting

Type	Instance	RAM
Master(m5.xlarge)	1	16GB
Core (m5.xlarge)	2	16GB

Learnings and Next Steps

- Reading hd5 files sequentially and writing to csv files
- Attaching EBS snapshot of 1 Million song (480GB)to EC2 instance and processing data
- Setting up EMR cluster on AWS

- Analytical and Modelling goals for next part:
  - Curate one million songs data
  - Understanding song quality metrics like time series of song loudness / song segment distribution
  - Build song recommender system like the artist recommendation data we have

Thank You