Spotify Recommender



By: Brendan Lo

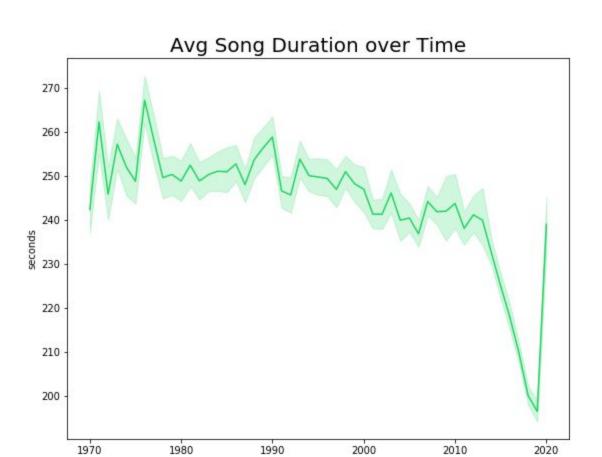
258 million active users

113 million subscribers

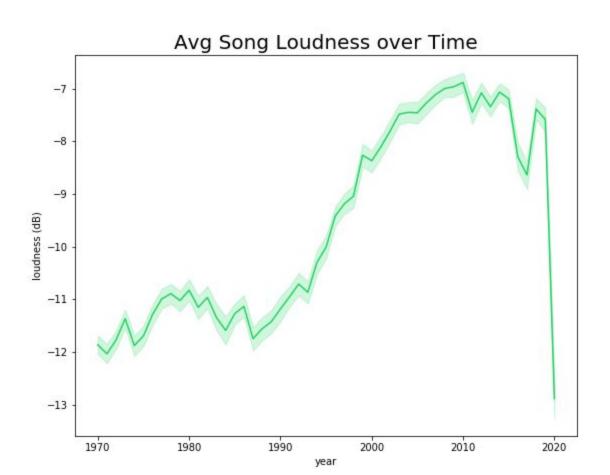
Getting Data from Spotify's Web API

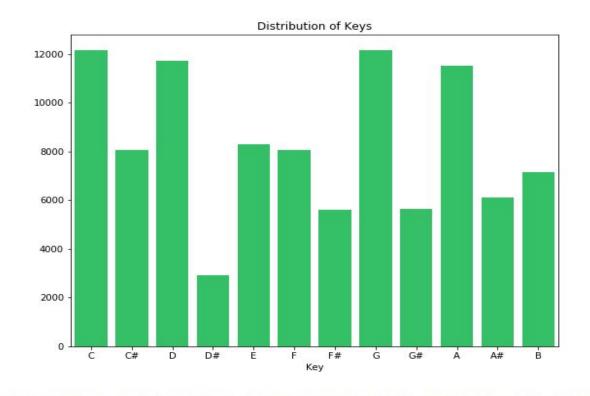
- Used Spotify's Web API spotipy
- Scraped 2,000 tracks per year between the years 1970-2020, resulting in about 100,000 tracks.
- 17 features. Metrics were <u>predefined by Spotify</u> based on their internal algorithms.

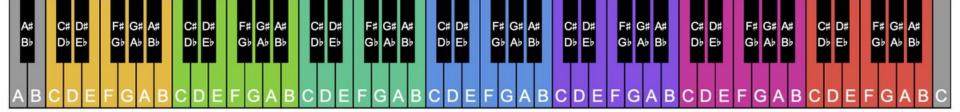
							Correl	ation He	atmap								
year -	- 1	0.0057	0.0048		0.009		0.66	0.29		0.13	0.12	-0.043	-0.061	0.15			
key -	0.0057	1	-0.00057		0.0053	-0.0022	0.0024	0.023	-0.026	0.026	0.034	-0.014	-0.0054	0.027	0.034		- 0.8
tempo -	0.0048	-0.00057	1	0.02	0.016	-0.04	0.0092	0.17			0.21		0.017	0.02	0.11		
mode ·	-0.068		0.02	1	-0.02	-0.055		-0.036	0.069		-0.062	-0.034	0.0085		0.0038		
time_signature -	0.009	0.0053	0.016	-0.02	1	0.00062	0.036	0.17		0.17	0.19		0.0014	0.019	0.13		- 0.4
duration_ms	-0.092	-0.0022	-0.04	-0.055	0.00062	1	-0.084	-0.065	-0.037		-0.013	0.11	0.049	-0.053			
popularity -	- 0.66	0.0024	0.0092		0.036		1	0.3	-0.13	0.18	0.12			0.081			
loudness -	0.29	0.023	0.17	-0.036	0.17		0.3	1	-0.58	0.26	0.76	-0.43	0.081	0.095	0.25		- 0.0
acousticness -	-0.11	-0.026		0.069		-0.037		-0.58	1		-0.72	0.24			-0.23		
danceability -	0.13	0.026			0.17		0.18	0.26	-0.2	1	0.13	-0.3		0.19	0.53		
energy -	0.12	0.034	0.21	-0.062	0.19	-0.013	0.12	0.76	-0.72	0.13	1	-0.24	0.18	0.14	0.34	3	-0.4
instrumentalness -	-0.043	-0.014		-0.034		0.11		-0.43	0.24	-0.3	-0.24	1	-0.039				
liveness -	-0.061	-0.0054	0.017	0.0085	0.0014	0.049		0.081			0.18	-0.039	1	0.16	-0.024		
speechiness -	0.15	0.027	0.02		0.019	-0.053	0.081	0.095		0.19	0.14		0.16	1	0.028		0.8
valence -	-0.17	0.034	0.11	0.0038	0.13	-0.16		0.25	-0.23	0.53	0.34	-0.23	-0.024	0.028	1		
	year -	key -	- odwaj	mode -	time_signature -	duration_ms -	popularity -	loudness -	acousticness -	danceability -	energy -	instrumentalness -	liveness -	speechiness -	valence -		

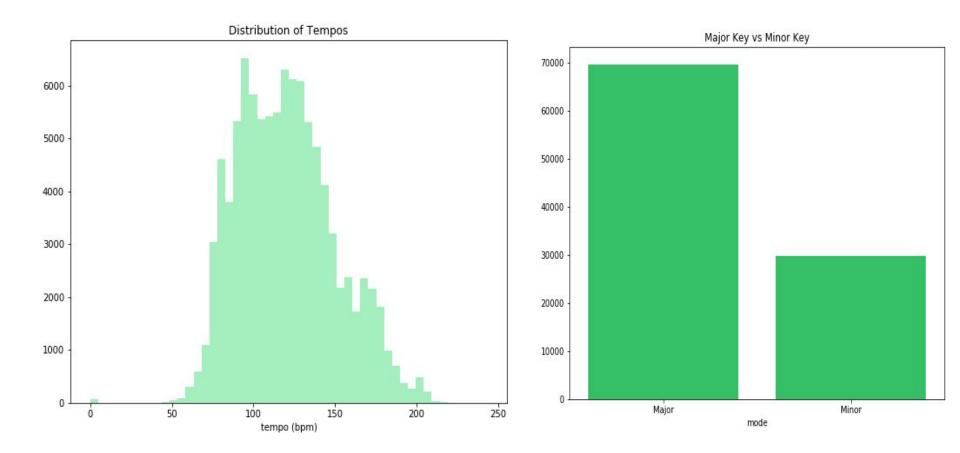


year







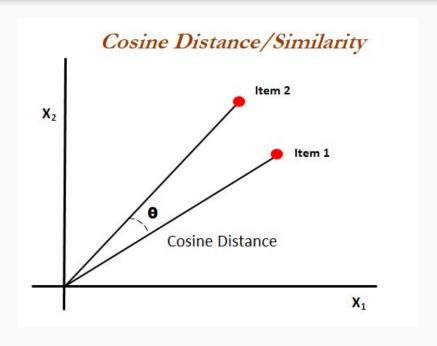


Preprocessing Concepts

- Standard Scaling
- Cosine Similarity
- Singular Value Decomposition (SVD)

Cosine Similarity

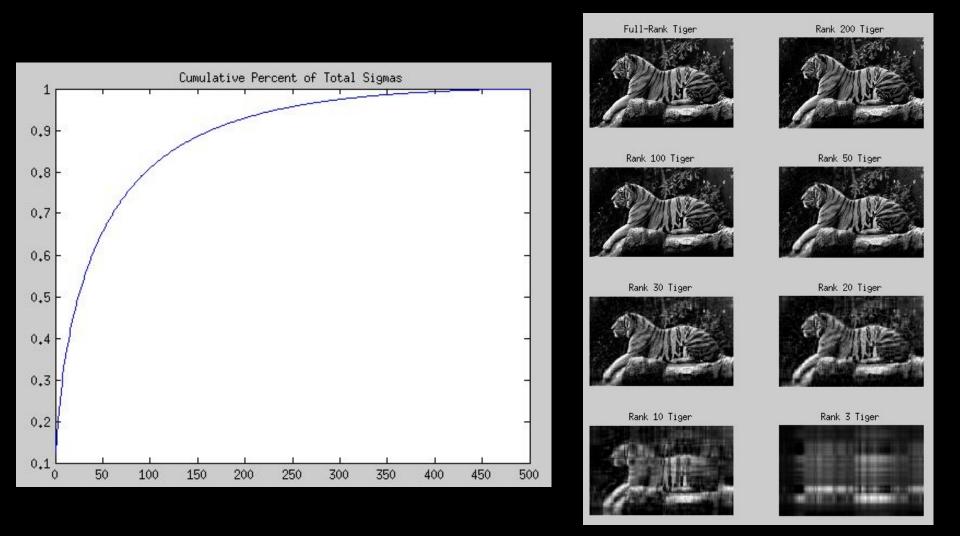
- Comparing the angle of vectors
- Lines overlap at 1
- Orthogonal lines are 0
- Lines in the opposite direction are -1
- In our case each song with its features define a single vector



Singular Value Decomposition (SVD)

- Factorizes a matrix M of any size into the product of 3 simpler matrixes.

- Useful for dimensionality reduction.



Product-based Recommender System

```
[17]: recommend1('closer, the chainsmokers')
closer, the chainsmokers
10 closest songs
track name
just the same, charlotte lawrence
                                                                               0.966189
fuck, i'm lonely (with anne-marie) - from "13 reasons why: season 3", lauv
                                                                               0.950468
mi verdad (feat. shakira), maná
                                                                               0.948197
fallin' all in you, shawn mendes
                                                                               0.937250
panama, quinn xcii
                                                                               0.936350
tough (feat. noah kahan), guinn xcii
                                                                               0.930817
no vuelvas más, darell
                                                                               0.923703
despacio, natti natasha
                                                                               0.921614
                                                                               0.915895
cool again, shoffy
bloom, black party
                                                                               0.912240
Name: closer, the chainsmokers, dtype: float64
```

Recommender Takeaways

- Recommender recommended songs based on the audio's features.
- This resulted in output songs that were musically similar, yet sound completely different
- Good for exploring new music OUTSIDE your comfort zone.

Limits and Next Steps...

- Need more memory and computing power. There are a lot more than 2,000 songs per year.
- Understand the data better. When I query 2,000 tracks per year, how does Spotify decide which tracks of that year?
- Improve recommender by incorporating user ratings/feedback.
- Create an app that is user-friendly, like with Flask.

Demo time!