IRON HACK

Spotify®

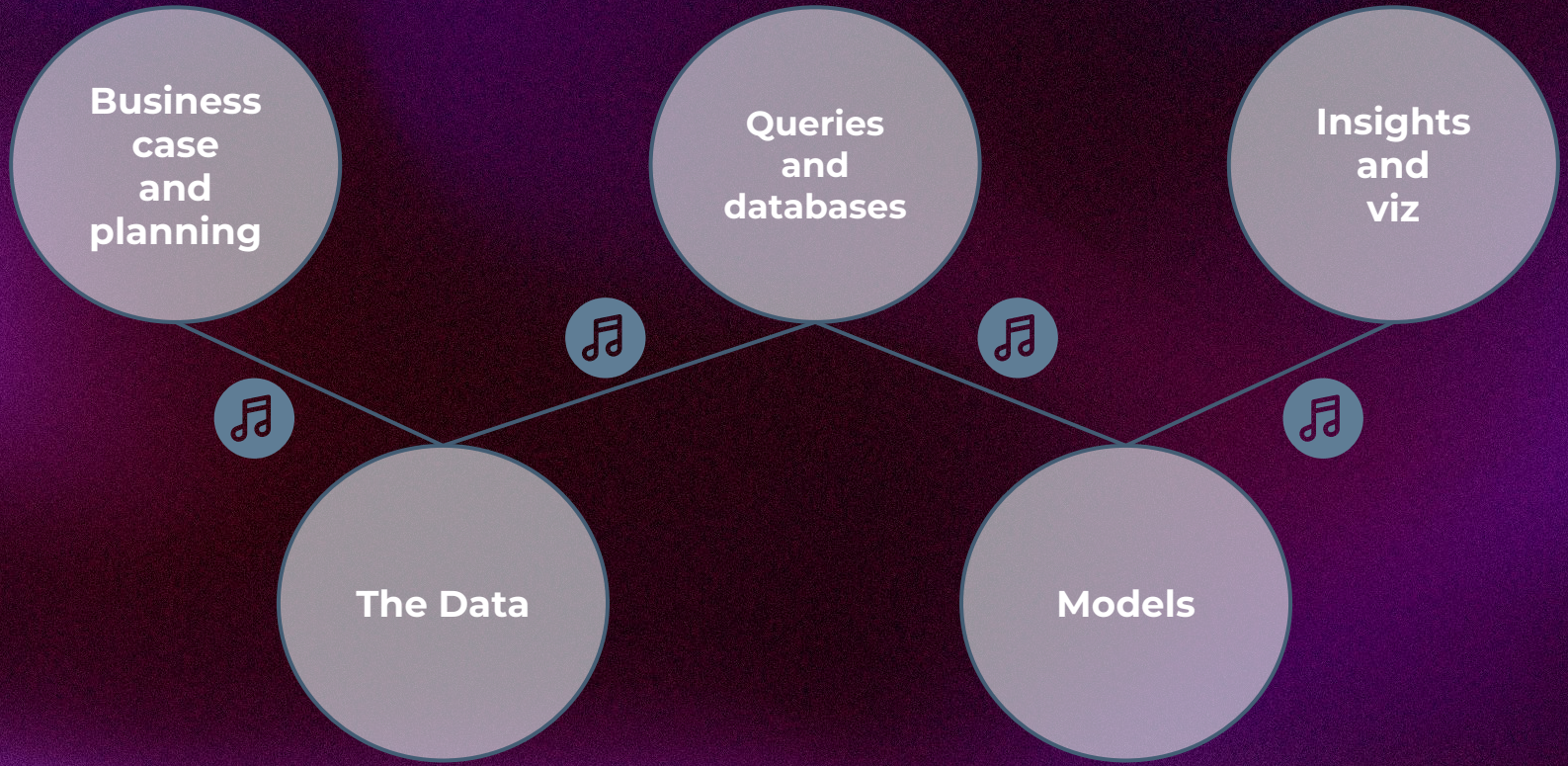
Historic and personal analysis of music based on Spotify.

Jaime Sastre Crespo

DATA Analytics 2022

DEC, 9th | PARIS

Contents of this presentation



Business case

#01

Music-lover

#02

Music evolution

#03

Spotify

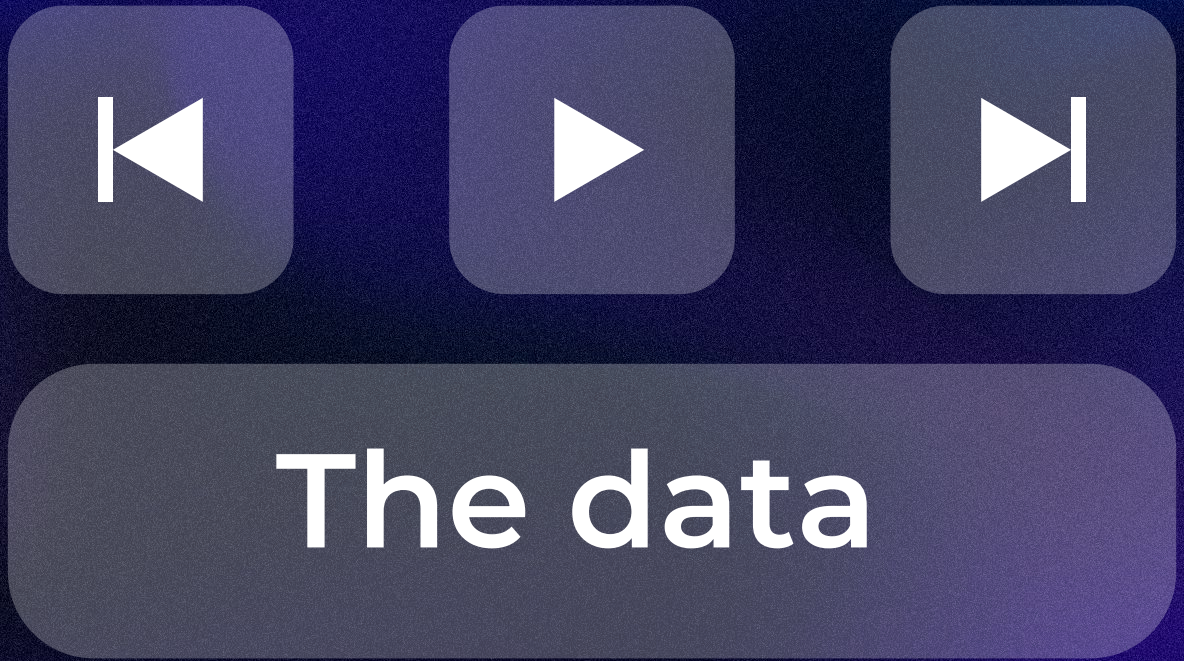
Planning

Data cleaning and preparation, EDA;

Databases, Queries, ERD;

Supervised ML, Model comparison;

Visualisations, results.


The data

#01

API Spotify

- Two ways:
  - Top 100 songs from 2018 to 2021
  - 150 currents songs
- 535 rows, 18 columns

#02

Spotify streaming

- 1 year of your historical data
- Your playlist of loved songs
- 23490 rows x 10 columns

#03

Historical data

- Kaggle data
- Songs from 1920 to 2020
- 586672 rows, 20 col

Data cleaning and preparation

Missing values / duplicated

Drop 60.000 rows duplicated in hist df.

```
duplicate_names_artist = df[df.duplicated(['name', 'artists'])]
```

Modify some columns :

- Convert column of milliseconds into minutes/seconds
- Change type of column into datetime to operate

Add some column for the EDA/Viz :

- Split the endTime column into hour, day and month
- UniqueID : Artist + track name

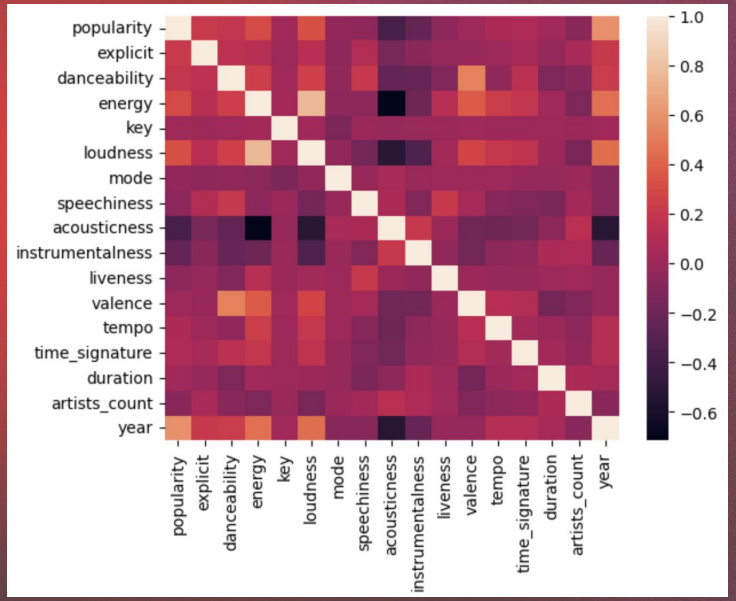
```
df_library['UniqueID'] = df_library['artist'] + ":" + df_library['track']
```

Create column InLibrary in my df of streaming.

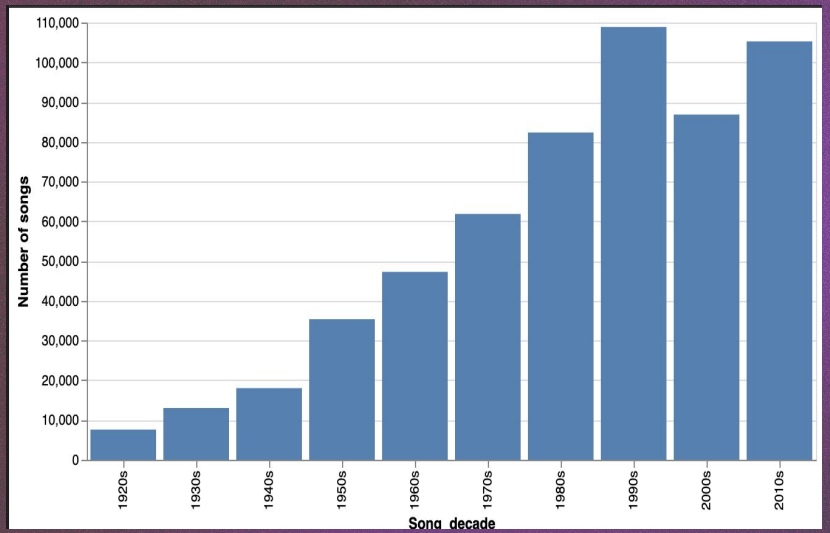
```
df_final['In Library'] = np.where(df_final['UniqueID'].isin(df_library['UniqueID'].tolist()),1,0)
```

Exploratory Data Analysis (EDA)

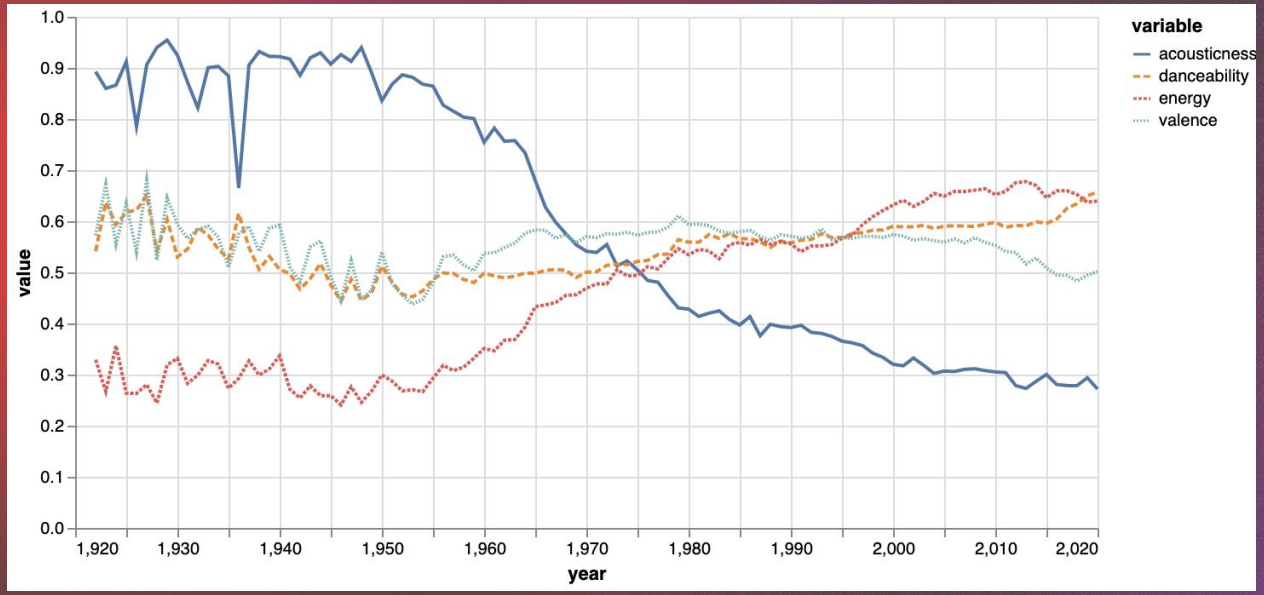
Correlation heat map of historical df:



Total number of song per decade (1920s to 2010s) - historical df:



Exploratory Data Analysis (EDA)



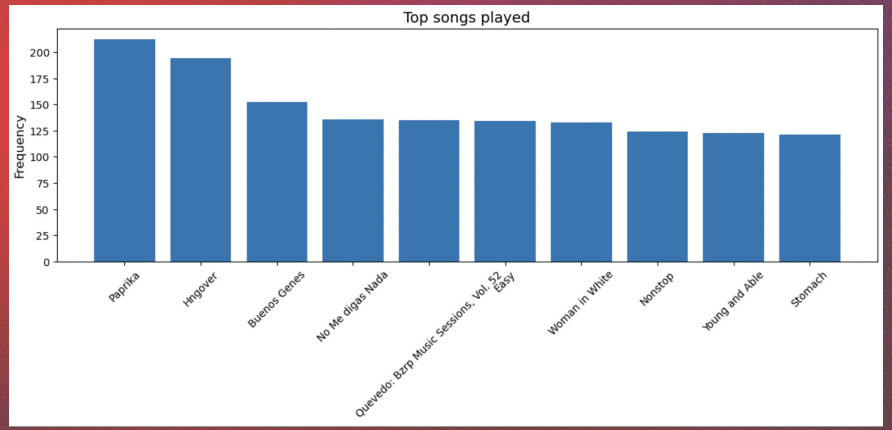
Evolution of some features over the years:

- Acousticness
- Danceability
- Energy
- Valence

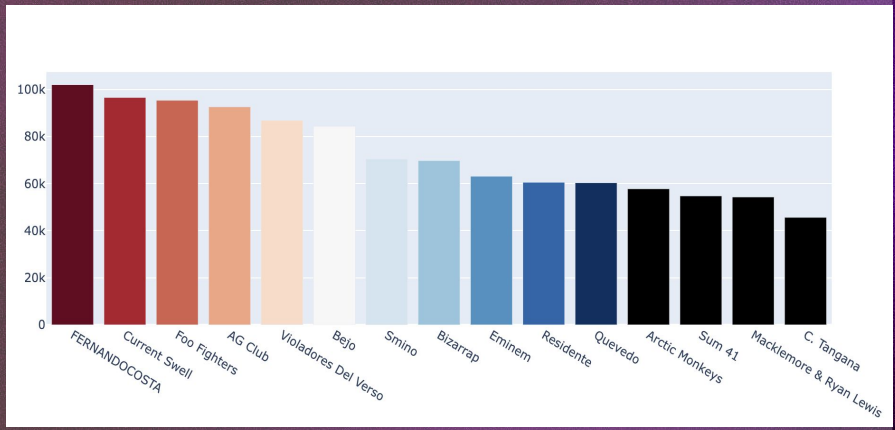
Exploratory Data Analysis (EDA)

Top songs played from my streaming data:

- Dec 21' to Dec 22'

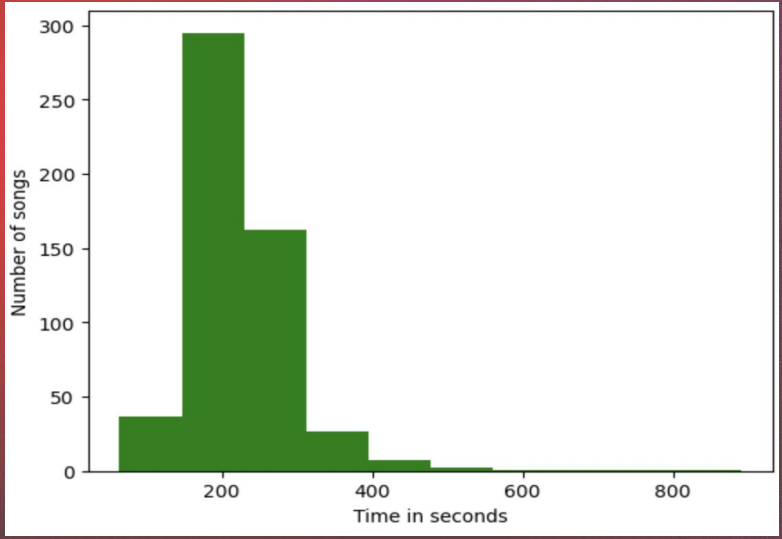


Top artists based in total of minutes listened from my streaming data:

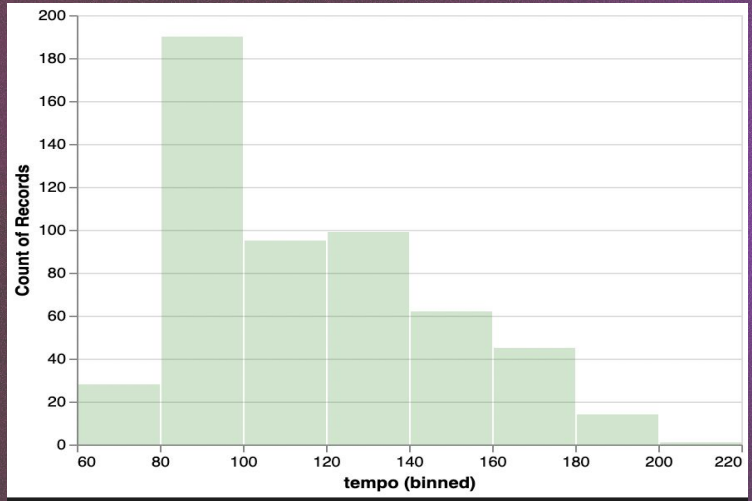


Exploratory Data Analysis (EDA)

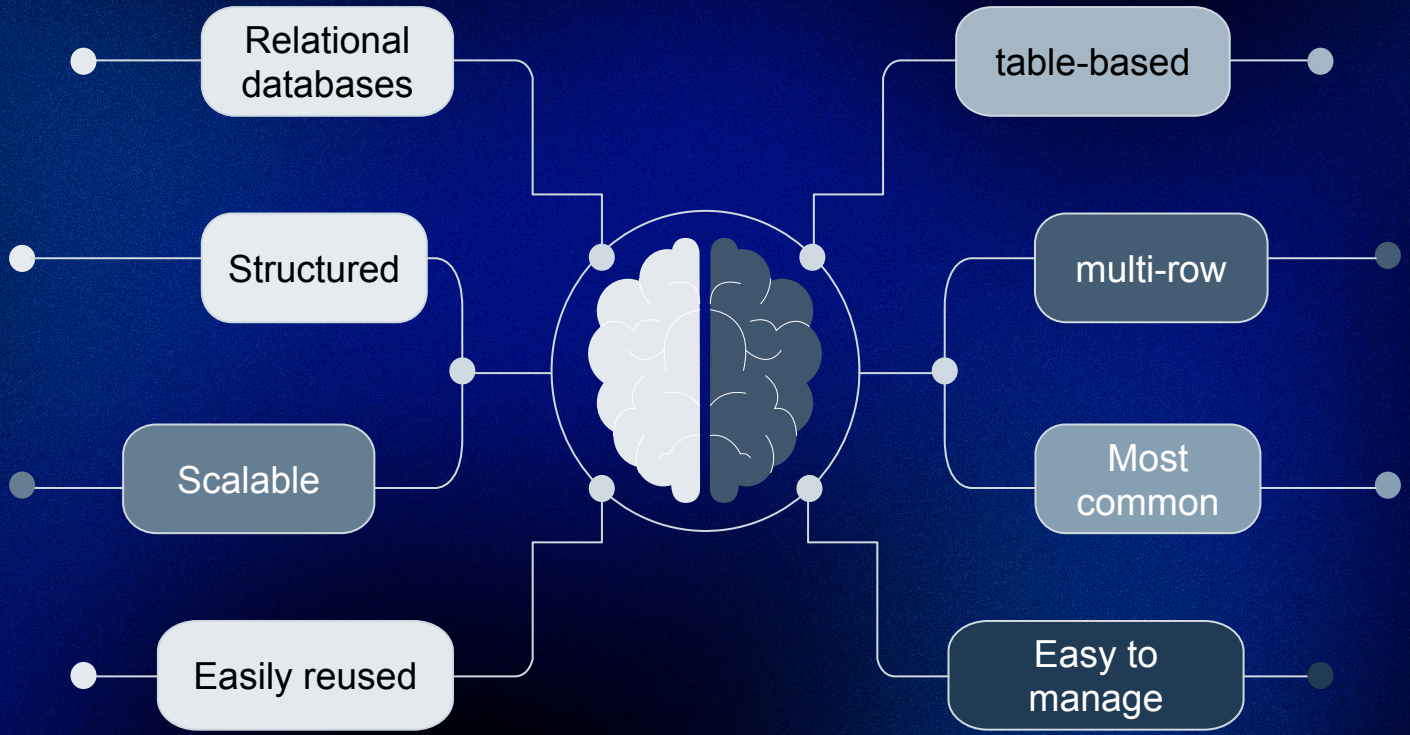
Number of songs grouped by time of the dataset of my top songs from 2018 to 2022:



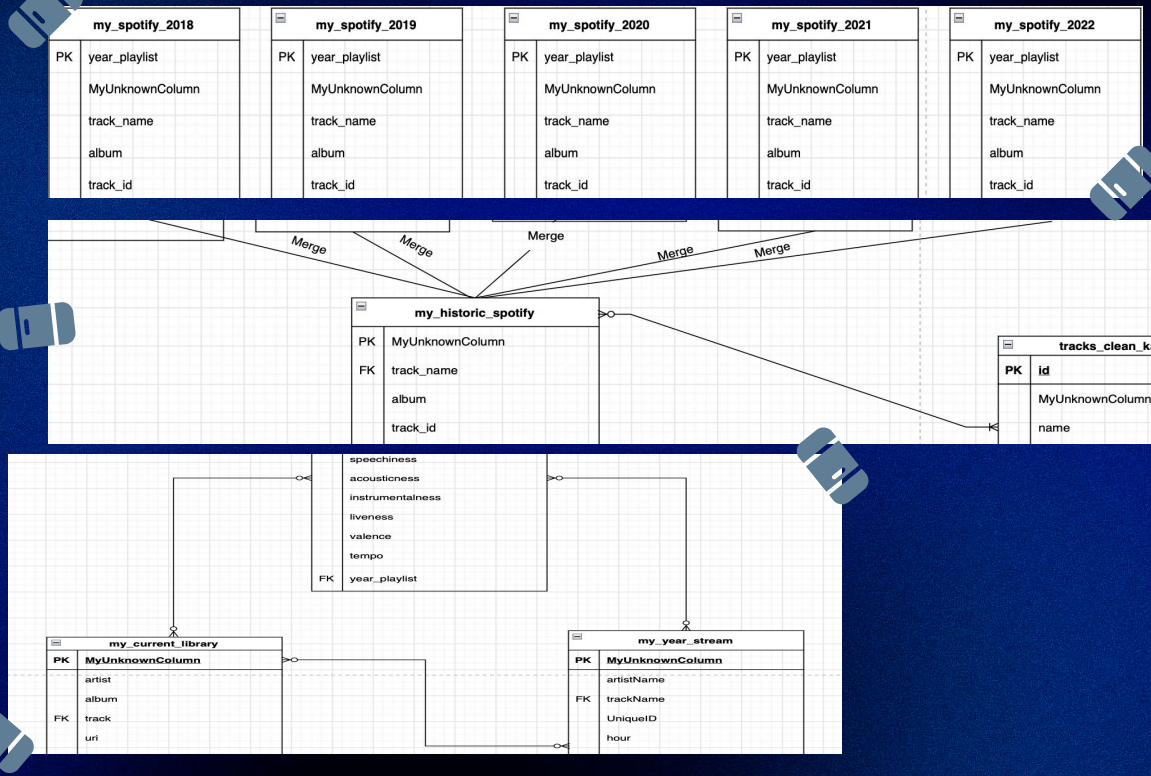
Distribution of songs by tempo:



Database selection: Why SQL?



Entity Relationship Diagram (ERD)



MySQL

```sql
CREATE DATABASE IF NOT EXISTS final_project;

USE final_project;

SELECT *
FROM my_year_stream;


ALTER TABLE my_current_library
ADD PRIMARY KEY (MyUnknownColumn);

```

wizard process to import databases

MySQL: queries


```
15    -- 1
16    -- To show the song and artist I played most during the year, how many times and how much time
17
18    SELECT count(artistName) as Max_artist, UniqueID, count(day), sum(duration_ms)
19    FROM my_year_stream
20    GROUP BY UniqueID
21    ORDER BY count(artistName) DESC ;
22
```

00% 13:20

Result Grid | Filter Rows: Search | Export: | Fetch rows:

Max_artist	UniqueID	count(da...	sum(duration_ms)
136	Quevedo:No Me digas Nada	136	18944
135	Bizarrap:Quevedo: Bzrp Music Sessions, Vol. 52	135	26056
133	Current Swell:Woman in White	133	21088
123	Quevedo:Nonstop	123	24428
123	Current Swell:Young and Able	123	26201
121	Current Swell:Stomach	121	22419
116	Eminem:Lose Yourself - From "8 Mile"" Soundtr…	116	28672
113	Young Thug:Livin It Up (with Post Malone & A$A…	113	21596
109	Ayax y Prok:Guajiro	109	14554
106	AG Club:Paprika	106	16455

MySQL: queries


```sql
-- 2
-- THE SONGS I PLAYED LESS THAN 2 SECS AND THEY ARE NOT IN MY CURRENT LIBRARY, MAYBE WE SHOULD REVIEW IF WE WANT TO KEEP THEM
-- IN OUR PLAYLIST (IF THEY ARE) OR IMPROVE THE MACHINE LEARNING RECOMMENDATION OF SPOTIFY TO AVOID THIS KIND OF MUSIC


SELECT count(artistName) as Counter,  mys.UniqueID
FROM my_year_stream mys
LEFT JOIN my_current_library mcl
ON mys.artistName = mcl.artist
WHERE duration_ms < 2 and mys.artistName NOT IN (SELECT artist FROM my_current_library)
GROUP BY mys.UniqueID
ORDER BY count(artistName) ASC
;
```

19:30

Result Grid | Filter Rows: Search | Export:

Counter	UniqueID
1	Oques Grasses:Inevitable
1	Els Pets:No vull que t'agradi aquesta canco
1	Manel:Benvolgut
1	Bongo Botrako:Revoltosa
1	Oques Grasses:Canco de l'aire
1	Els Pets:Blue tack
1	Manel:Sabotatge
1	Manel:Ai, Dolors
1	Cesk Freixas:La Petita Rambla del Poble Sec - XL
1	Oques Grasses:Serem ocells

MySQL: queries


```
38    -- 3
39    -- When did I listen more songs during the week last year? Display the day and the artist.
40
41    SELECT count(artistName), weekday
42    FROM  my_year_stream
43    GROUP BY weekday
44    ORDER BY count(artistName) DESC
45    LIMIT 3;
46
```

00% 11:42

Result Grid | Filter Rows: Search | Export: | Fetch rows:

count(artistNa...	weekday
3669	Saturday
3653	Friday
3393	Tuesday

MySQL: queries


```
48    -- 4
49    -- Show if there's any song (and with some level of popularity) that I did not listen in the last year but
50    -- it is in my historic spotify that records my top 100 from each of the following years: 2018, 2019, 2020,
51    -- 2021 and 2022 (150 songs of this last year)
52
53
54    SELECT DISTINCT mhs.artist,  mhs.track_name, popularity
55    FROM my_historic_spotify mhs
56    LEFT JOIN my_year_stream mys
57    on  mhs.track_name = mys.trackName
58    WHERE mhs.track_name NOT IN (SELECT trackName FROM my_year_stream)
59    AND popularity > 40;
60
```

100% 17:57

Result Grid | Filter Rows: Search | Export:

artist	track_name	popularity
Funambulista	Ya Veras (with Andres Suarez)	45
DELLAFUENTE	Guerrera	61
XXXTENTACION	SAD!	83
XXXTENTACION	the remedy for a broken heart (why am I so in lo…	79
Funambulista	Me Inventare (with Dani Martin)	43
La Gossa Sorda	Esbarzers	44
XXXTENTACION	Moonlight	82
El Canto Del Loco	Zapatillas	63
Wiz Khalifa	Gin and Drugs (feat. Problem)	45
Gritando en Silencio	A la luz de una sonrisa	42

MySQL: queries


```
62    -- 5
63    -- How Was my music in terms of matching popularity in the last 5 years?
64    SELECT track_name, artist , tempo , popularity, year_playlist
65    FROM my_historic_spotify
66    WHERE popularity > 75
67    ;
```

100% 15:65

Result Grid | Filter Rows: Search | Export:

track_name	artist	tempo	popularity	year_playlist
SAD!	XXXTENTACION	75.023	83	2018
the remedy for a broken heart (why am I so in lo…	XXXTENTACION	119.705	79	2018
R U Mine?	Arctic Monkeys	97.094	82	2018
Moonlight	XXXTENTACION	128.009	82	2018
Do I Wanna Know?	Arctic Monkeys	85.03	87	2018
God's Plan	Drake	77.169	84	2019
Last Resort	Papa Roach	90.598	80	2019
Demons	Imagine Dragons	89.938	84	2020
Natural	Imagine Dragons	100	81	2020
Thunder	Imagine Dragons	167.997	84	2020
Take Me To Church	Hozier	128.945	83	2020
Believer	Imagine Dragons	124.949	88	2020
Self Care	Mac Miller	141.894	79	2020
Weekend (feat. Miguel)	Mac Miller	120.058	77	2020
Little Talks	Of Monsters an…	102.961	79	2020
Demasiadas Mujeres	C. Tangana	126.043	79	2021
Mon Amour - Remix	zzoilo	116.041	83	2021
WITHOUT YOU	The Kid LAROI	93.005	79	2021
Small Worlds	Mac Miller	78.267	77	2021
The Scientist	Coldplay	146.277	85	2021
Everlong	Foo Fighters	158.066	83	2022
No Me digas Nada	Quevedo	102.16	78	2022
Ahora y Siempre	Quevedo	118.964	76	2022
Learn to Fly	Foo Fighters	135.997	76	2022
Smells Like Teen Spirit	Nirvana	116.761	82	2022
The Pretender	Foo Fighters	172.984	78	2022
I Like You (A Happier Song) (with Doja Cat)	Post Malone	100.964	89	2022
Quevedo: Bzrp Music Sessions, Vol. 52	Bizarrap	128.033	97	2022

Supervised ML models

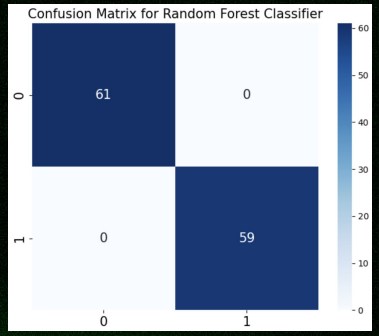
Model for predicting popularity

#01	Define features	X: songs features Y: popularity
#02	Undersampling	`from sklearn.datasets import make_classification` `from imblearn.under_sampling import ClusterCentroids`
#03	Split data in train and test	`X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, train_size=0.70, random_state=8)`
#04	TPOT	intelligently exploring thousands of possible pipelines to find the best one
#05	RandomForestClassifier/KNN/ BaggingClassifier	Saturn is composed of hydrogen and helium

Comparison between models

RFC

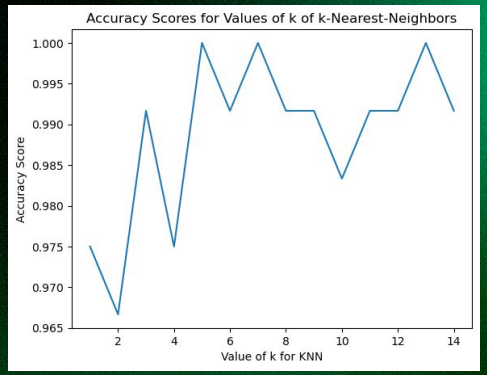
Precision: 100%
Accuracy: 100%
Recall: 100%
AUC: 100%



KNN

Precision: 100%
Accuracy: 99%
Recall: 100%
AUC: 100%

For k = 5



BC

Model test Score: 0.908,
Model training Score: 0.975

estimators: 20

Some more insights...

Recommendation playlist

Auth:

```
with open(path / "client_s.txt") as f:
    content = f.readlines()
content = [x.strip() for x in content]


client_id = content[0]
client_secret = content[1]
redirect_uri =
'http://127.0.0.1:8080/yep'
username = content[3]


client_credentials_manager = SpotifyClientCredentials(client_id=client_id,
                                                      client_secret=client_secret)

sp = spotipy.Spotify(client_credentials_manager = client_credentials_manager)
```

Seeds:

```
top_tracks_short = sp.current_user_top_tracks(limit = 50,offset=0,time_range='short_term')
top_tracks_med = sp.current_user_top_tracks(limit = 50,offset=0,time_range='medium_term')
top_tracks_long = sp.current_user_top_tracks(limit = 50,offset=0,time_range='long_term')
```

Recommendation playlist

Creating list of tracks recommended based on my last songs:

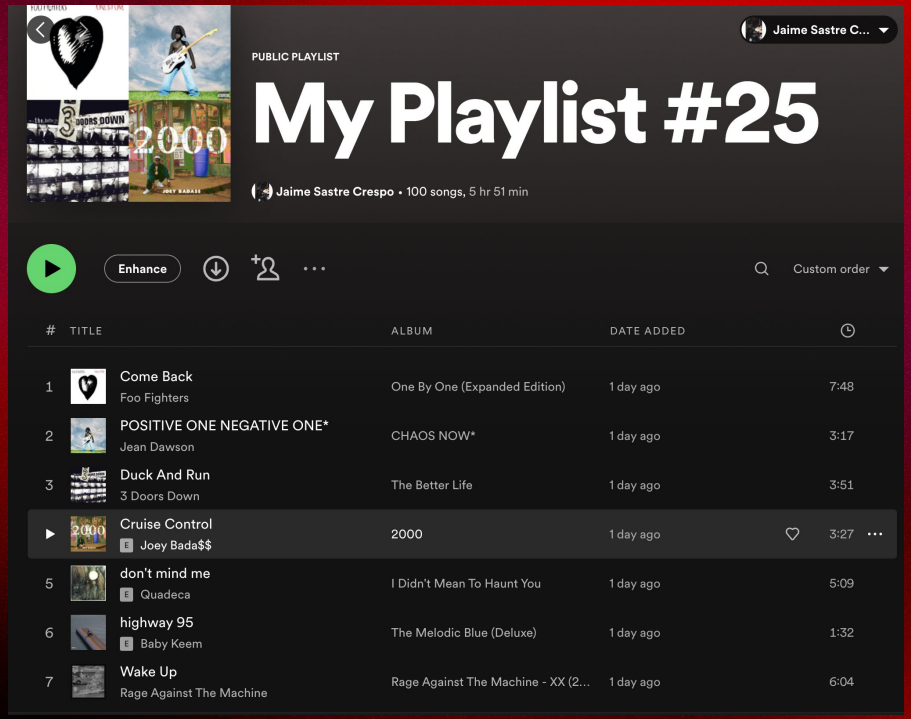
```
recomm_dfs = []
for i in range(5,len(seed_tracks)+1,5):
    recomms = sp_m.recommendations(seed_tracks = seed_tracks[i-5:i],limit = 10)
    recomms_df = append_audio_features(create_df_recommendations(recomms),sp_m)
    recomm_dfs.append(recomms_df)
recomms_df = pd.concat(recomm_dfs)
recomms_df.reset_index(drop = True, inplace = True)
```

Adding to a playlist created in my Spotify:

```
sp_m.user_playlist_add_tracks(username,

playlist_id="spotify:playlist:3XWyiIO6rtmYiJuW3Z91gy",
                              tracks = recomms_df["track_id"].tolist()[:100])
```

Playlist created



Challenges

Use Spotify API

Machine learning: computationally expensive

Work with different databases

Conclusions

#01	Music has bee losing accoustincness and gaining energy and power over years
#02	Gold decade during the 80s-90s
#03	Spotify and its API: huge application
#04	Spotify: recommendation to remove songs
#05	20 % of the music I listened is not in my library

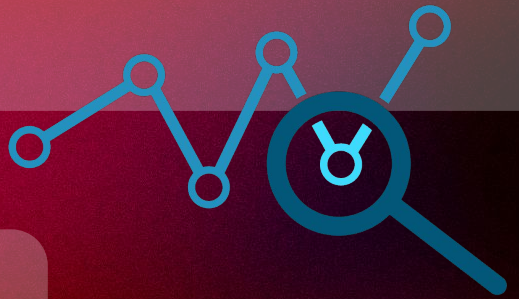
NEXT STEPS:

Make interactive the APP to create a playlist and more user friendly

Download on the App Store

Google play

Go further in the analysis



Deeper knowledge of the Spotify API

Thanks!

Do you have any questions?

Jaime Sastre Crespo
sastrecrespo@gmail.com
+34 669 92 74 94
https://github.com/JaimeSastreCrespo