Hit Music Prediction Model

Part 2

Agenda

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment

Business Understanding

What Is a Music Consultant?

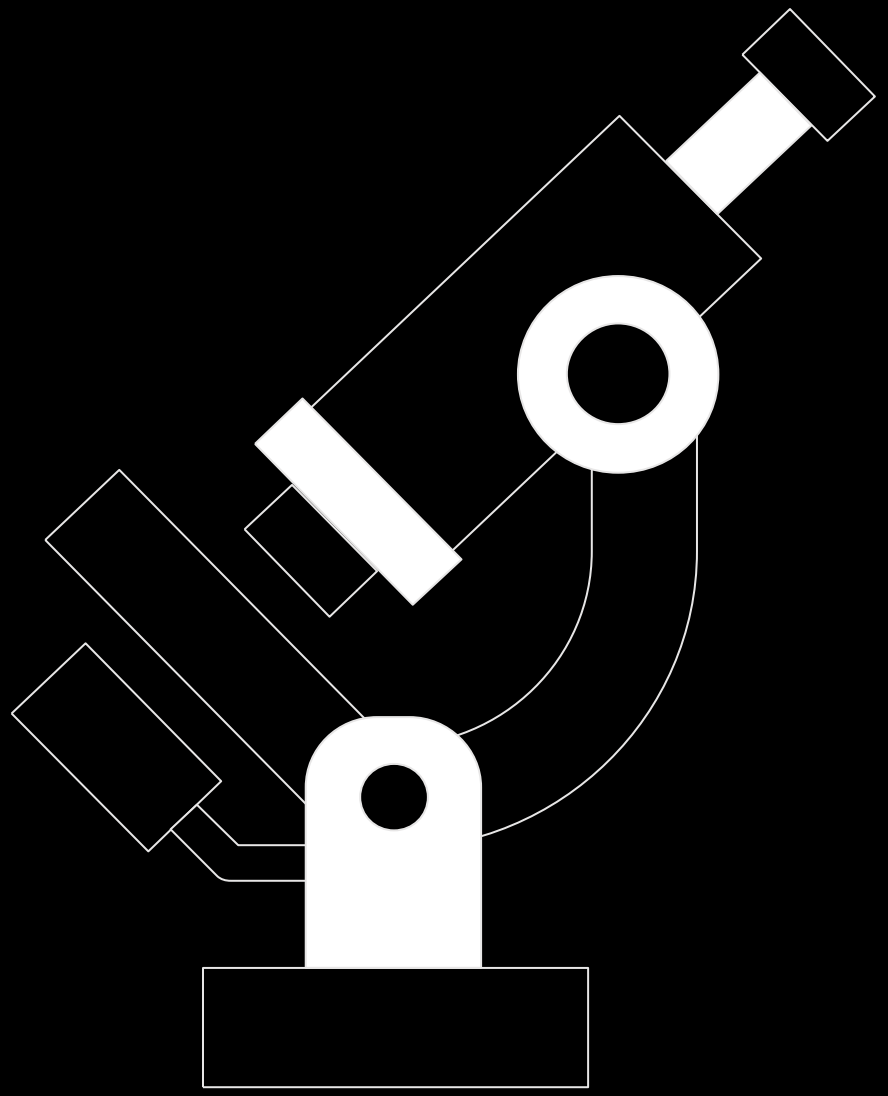
They guides the musicians/producers/ composers:

- Perform the research and studies to identify the art of what good music is
- They have a thorough idea of how music should be produced
- Composition of music

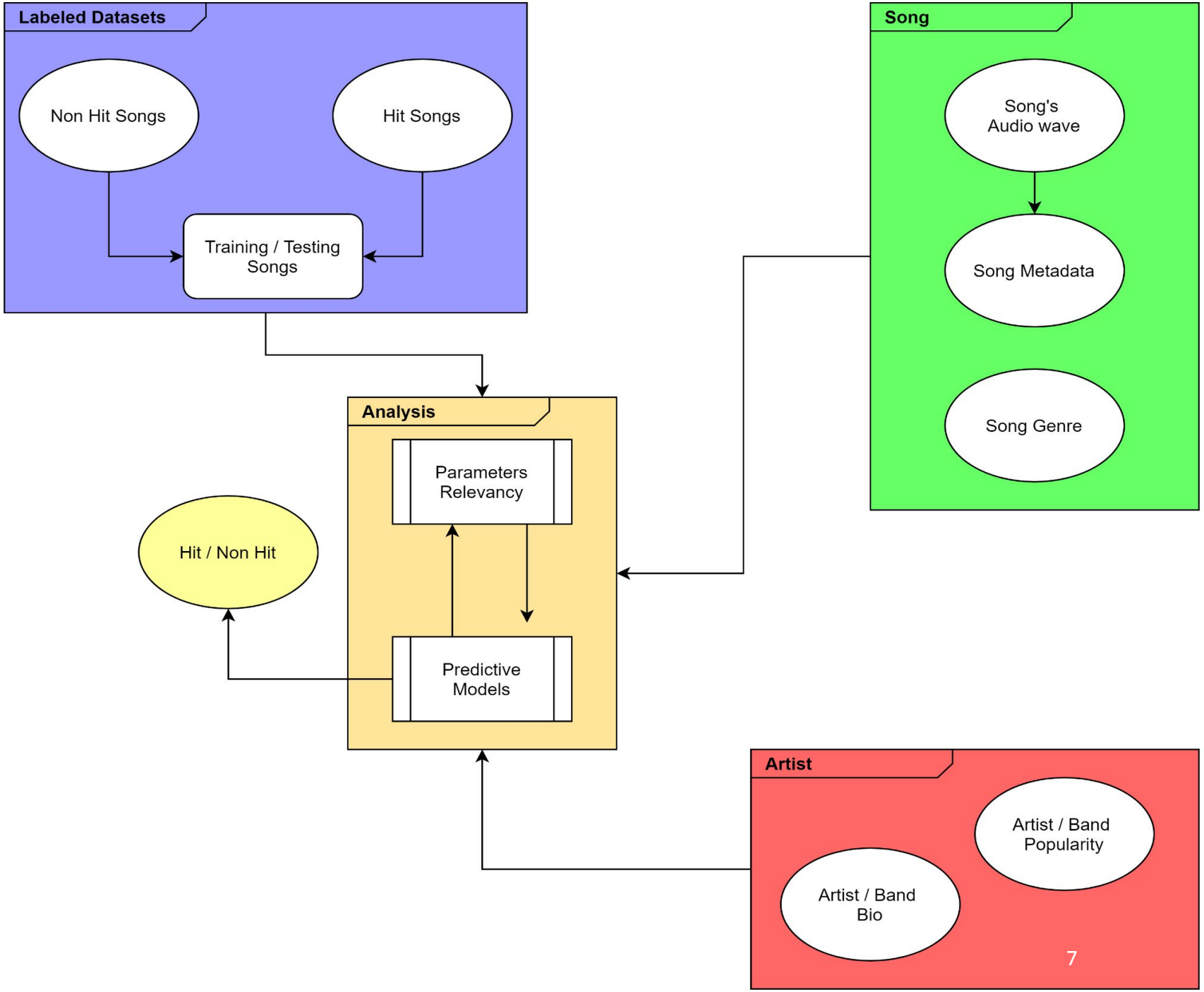
Our Product Will Be Addressing The Following Segmentations:

- Assists Record-Labels to identify upcoming artists/bands
- Help producers and composers create music based on data
- Major streaming services to quickly identify possible hit songs and direct their ads accordingly

Data Understanding



The first rough outline on how we could predict a song's success, looked like this



Command took 0.33 seconds -- by joseph.higaki@gmail.com at 3/24/2021, 8:48:14 AM on another test cluster

Cmd 6

Initialize Client Proxy

We can use sp, as the object that proxies our python code to the spotify api

This is where you paste your client id and secret from the application you created at the spotify developer site

Cmd 7

```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials

sp = spotipy.Spotify(auth_manager=SpotifyClientCredentials(client_id="client_id",
                                                           client_secret="client_secret"))
```

Command took 0.13 seconds -- by joseph.higaki@gmail.com at 5/1/2021, 7:26:42 PM on Eldery Cluster Behind the Counter in a small town

Cmd 8

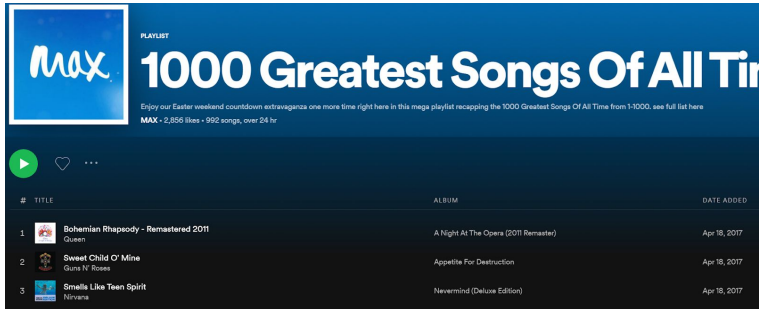
Get Audio Features

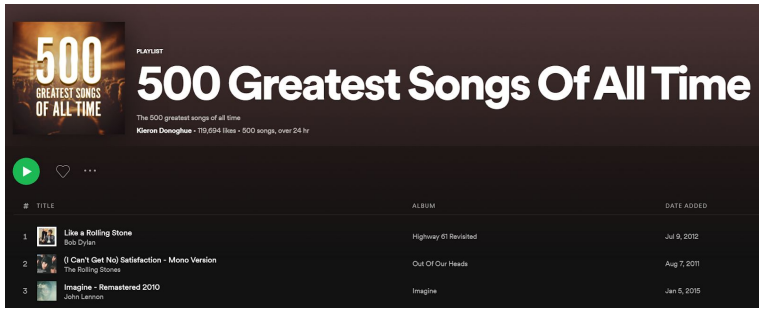
Multiple track ids can be used to query

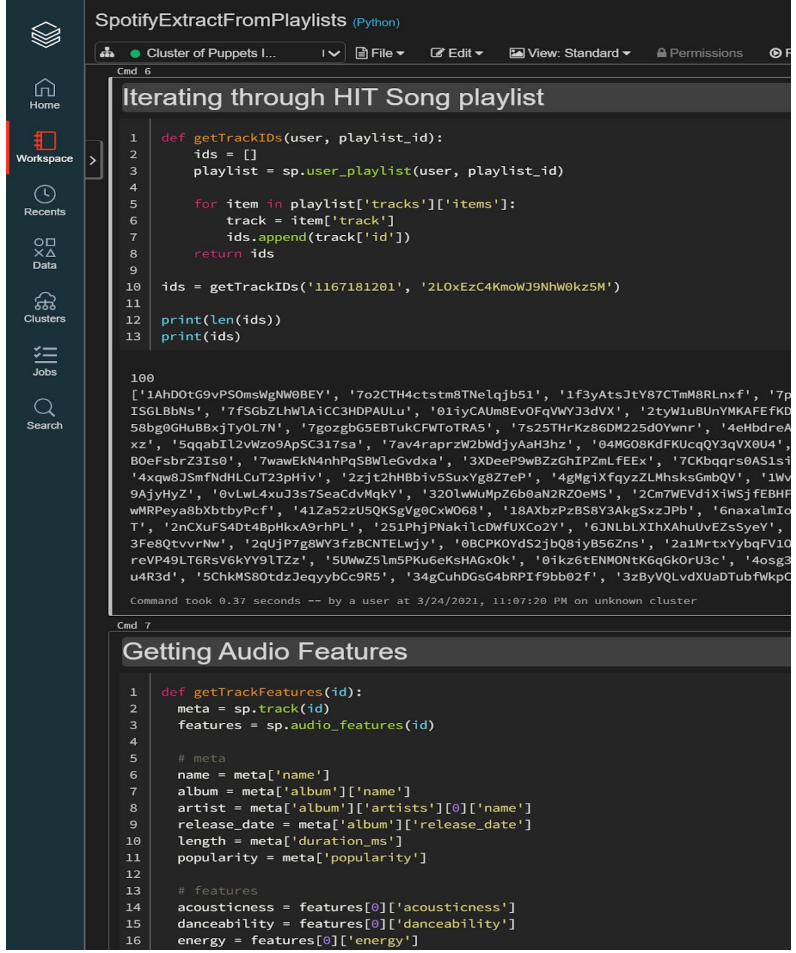
This is based on the API https://developer.spotify.com/documentation/web-api/reference/#endpoint-get-several-audio-features

THe response object and attributes are explained at https://developer.spotify.com/documentation/web-api/reference/#object-audiofeaturesobject

SPOTIFY API







SPOTIFY PLAYLISTS

kworb.net/spotify/country/global_weekly_totals.html

ITUNES	ITUNES WW	ARTISTS	US SALES	FOC
RADIO	SPOTIFY	YOUTUBE	CHARTS	HO

Main Page | All Artists

Spotify Weekly Chart Totals - Global | Click the headers to sort | Back to ch

Covers charts from 2013/09/29 to 2021/04/08.
Totals do not include time spent outside the weekly chart.

Pos	Artist and Title	Wks	T10	Pk (x?)	PkStreams	Total
1	Ed Sheeran - Shape of You	219	34	1 (x14)	64,217,796	2,732,556,879
2	Tones and I - Dance Monkey	90	41	1 (x17)	52,055,226	2,126,230,962
3	The Weeknd - Blinding Lights	71	64	1 (x13)	52,375,259	2,092,241,811
4	Post Malone - rockstar	172	27	1 (x17)	46,995,997	2,072,812,831
5	Post Malone - Sunflower - Spider-Man: Into the ...	129	31	1 (x2)	34,579,416	1,836,742,233
6	Lewis Capaldi - Someone You Loved	118	15	4	24,962,682	1,805,676,498
7	Shawn Mendes - Señorita	94	[illegible]	[illegible]	[illegible]37,638	1,766,337,672
8	Billie Eilish - bad guy	106	24	1 (x8)	30,342,324	1,724,472,059
9	The Chainsmokers - Closer	184	28	1 (x11)	46,300,740	1,705,086,994
10	James Arthur - Say You Won't Let Go	235	15	7	19,297,939	1,654,415,540
11	Imagine Dragons - Believer	218		16	16,864,827	1,652,573,414

view-source

```
34 <thead><tr><th class="np
35 <tr class="d1"><td class=
36 <td class="text mp"><div
37 <td>219</td>
38 <td>34</td>
39 <td>1</td><td class="np m
40 <td>64,217,796</td>
41 <td>2,732,556,879</td></t
42 <tr class="d0"><td class=
43 <td class="text mp"><div
44 <td>90</td>
45 <td>41</td>
46 <td>1</td><td class="np m
47 <td>52,055,226</td>
48 <td>2,126,230,962</td></t
49 <tr class="d1"><td class=
50 <td class="text mp"><div
51 <td>71</td>
52 <td>64</td>
53 <td>1</td><td class="np m
54 <td>52,375,259</td>
55 <td>2,092,241,811</td></t
56 <tr class="d0"><td class=
57 <td class="text mp"><div
58 <td>172</td>
59 <td>27</td>
60 <td>1</td><td class="np m
61 <td>46,995,997</td>
62 <td>2,072,812,831</td></t
```

KWORB

ARTISTS METADATA DATABASES

HypeAuditor DASHBOARD

Top Music Influencers On Instagram

by the number of quality and engaged followers

Find top Music influencers on Instagram to make the most of your marketing campaigns. We rank Instagram accounts by quality audience and authentic engagement. This makes our rank different.

April 2021 | Music | All Countries | How it works? | Share

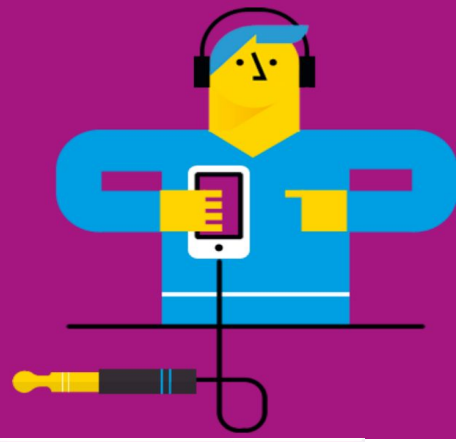
		Influencer	Category	Followe	Audience c	Authenti	En
1		arianagrande Ariana Grande	Music	232.6M	United States	2.9M	4.
2	3	billieeilish BILLIE EILISH	Music	82.3M	United States	8.7M	11.
3	4	selenagomez Selena Gomez	Music Lifestyle	222.6M	Azerbaijan	3.2M	4.
4	1	beyonce Beyoncé	Music Fashion	174M	United States	1.8M	2.

music story

OFFER LABELS USE CASES ABOUT US EN FR

We create **bonds** between **people and music**

Music Story is a metadata provider fully dedicated to music. We improve your digital music service with the most qualitative music metadata offer on the market.



MusicBrainz

Log In Create Account Search Artist

About Us Products Search Documentation

Queen (UK rock group)
~ Group

Overview | Releases | Recordings | Works | Events | Relationships | Aliases | Tags | Details | Edit

Annotation

Note: Please add any 2011 Remaster releases to either *Universal Island Records* or *Hollywood Records* and **not *Island Records***. Any previously added releases which fall under this category should be edited.

Annotation last modified on 2012-09-25 22:21 UTC.

Wikipedia

Queen are a British rock band formed in London in 1970. Their classic line-up was Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass). Their earliest works were influenced by progressive rock, hard rock and heavy metal, but the band gradually ventured into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock.

Show more...

Continue reading at Wikipedia... Wikipedia content provided under the terms of the Creative Commons BY-SA license

Discography

Filter

Previous 1 2 Next

Artist information
Type: Group
Founded: 1970-06-27 (50
Founded in: London, Engl
United Kingdom
Area: United Kingdom
ISNI code: 0000 0001 15

Rating
(see all ratings)

Genres
rock, hard rock, glam rock, pop rock

Other tags
british, awesome, uk, 70's,
See all tags

External links
Official homepage
@officialqueenmusic
@QueenWillRock
45cat
AllMusic
BnF Catalogue
Discogs
DNB

music_story.py × | ig_search_test.py | instagram.py | decorate_artist_list.py | UNION MSI01-IS01-IS02-IS03-IS04-IS05 - Not Found - Removed D

.vscode > music_story.py > MusicStory > _extract_music_story_artist_id

```python
        ):
        # Extract URL value from MusicStory response XML document
        return MusicStory._extract_music_story_data_item_first_node_value(xml_document_text, "id")

    @staticmethod
    def _extract_music_story_artist_url(
        xml_document_text
        ):
        # Extract URL value from MusicStory response XML document
        return MusicStory._extract_music_story_data_item_first_node_value(xml_document_text, "url")

    @staticmethod
    def _extract_username_from_site_url(site_url):
        # Extract first path element after"*.com" like url. Can be used to extract instagram username
        return site_url.rsplit('.com/', 1)[-1].rstrip("/")

    def get_music_story_artist(self, spotify_artist_id):
        # Queries the Music Story Artist using Spotify artist id
        # Adds the artist instagram url and username, according to the Music-Story database
        # Returns a Music Story Artist object
        # TO-DO: extract facebook, twitter and wikimedia links
        music_story_artist = MusicStoryArtist()
        music_story_artist.id = MusicStory._extract_music_story_artist_id(
            self._get_music_story_artist_by_spotify_artist_id(spotify_artist_id = spotify_artist_id))
        instagram_response = self.get_music_story_instagram_by_id(music_story_artist.id)
        music_story_artist.instagram_url = MusicStory._extract_music_story_artist_url(instagram_response)
        music_story_artist.instagram_username = MusicStory._extract_username_from_site_url(music_story_artist.instagram_url)
        return music_story_artist
```

Instagram

INSTAGRAM NAME SEARCH

instagram.py × | ig_search_cranberries.json | JsonViewer × | decorate_artist_list.py | UNION MSI01-ISO...

object ▶ users ▶ 1 ▶ user ▶


```
▼ object {7}
  ▼ users [43]
    ▼ 0 {2}
        position : 0
      ▼ user {12}
          pk : 13433161
          username : thecranberries
          full_name : The Cranberries
          is_private : false
          profile_pic_url : https://scontent-mad1-1.cdninstagram.com/v/t51.2885-19/s150x150/81777923_120111
                            1.cdninstagram.com&_nc_ohc=EW0fqdYyHZIAX8mX9P8&edm=AHG7ALcBAAAA&ccb=7-4&oh=
          profile_pic_id : 2230172204060916975_13433161
          is_verified : true
          has_anonymous_profile_picture : false
        ▶ account_badges [0]
        ▶ friendship_status {6}
```

object ▶ graphql ▶

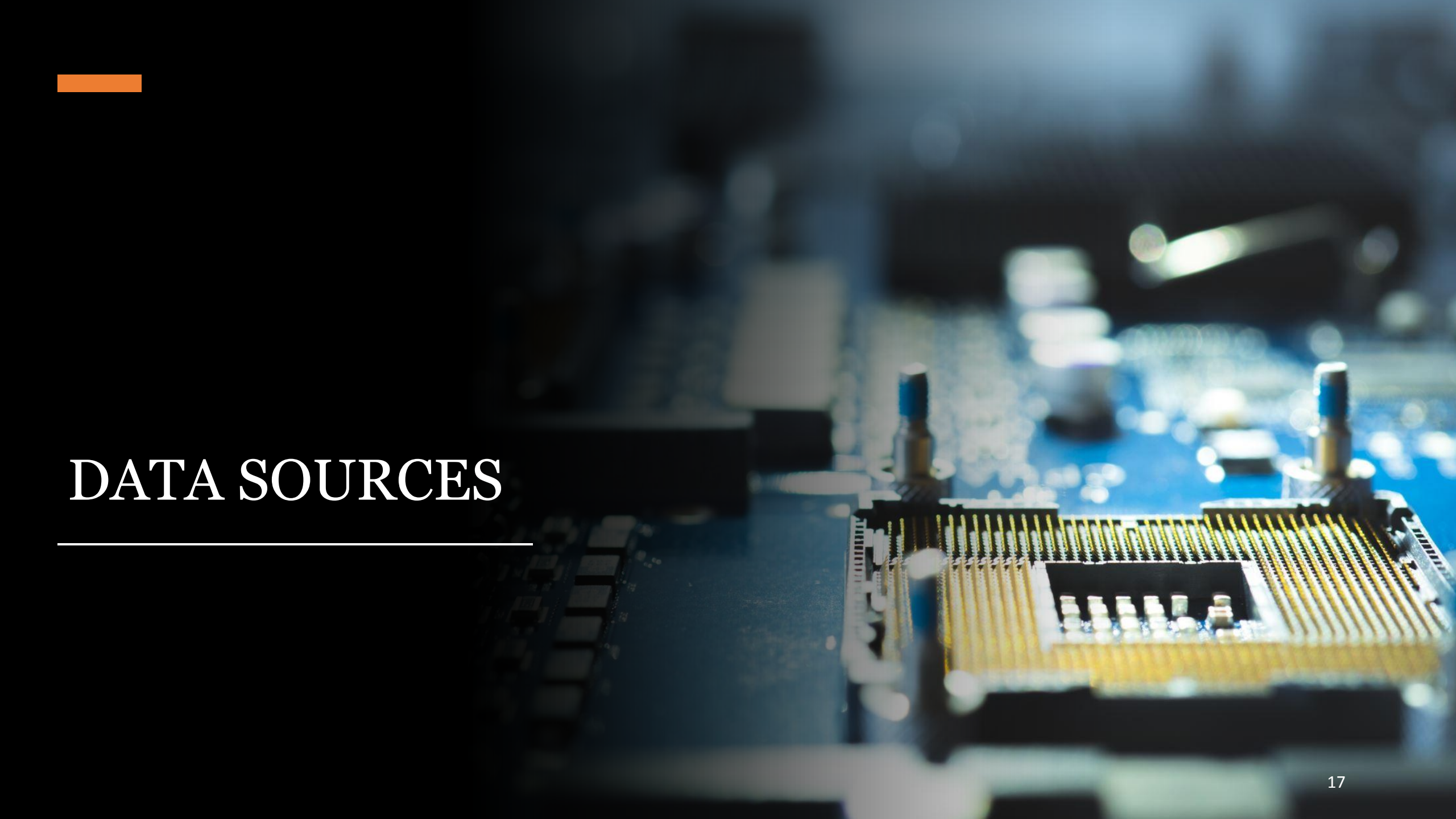
```
▼ object {8}
    logging_page_id : profilePage_13433161
    show_suggested_profiles : false
    show_follow_dialog : false
  ▼ graphql {1}
    ▼ user {41}
        biography : The official Instagram account for The Cranberries. ‘Yesterday’s Gone’ i
        blocked_by_viewer : false
        restricted_by_viewer : false
        country_block : false
        external_url : https://thecranberries.lnk.to/IGCloseToYou
        external_url_linkshimmed : https://l.instagram.com/?u=https%3A%2F%2Fthecranberries.l
                                   EvlnLpmMupdxpQ&s=1
      ▼ edge_followed_by {1}
          count : 176364
        fbid : 17841401714830209
        followed_by_viewer : true
      ▼ edge_follow {1}
          count : 3
        follows_viewer : false
        full_name : The Cranberries
```

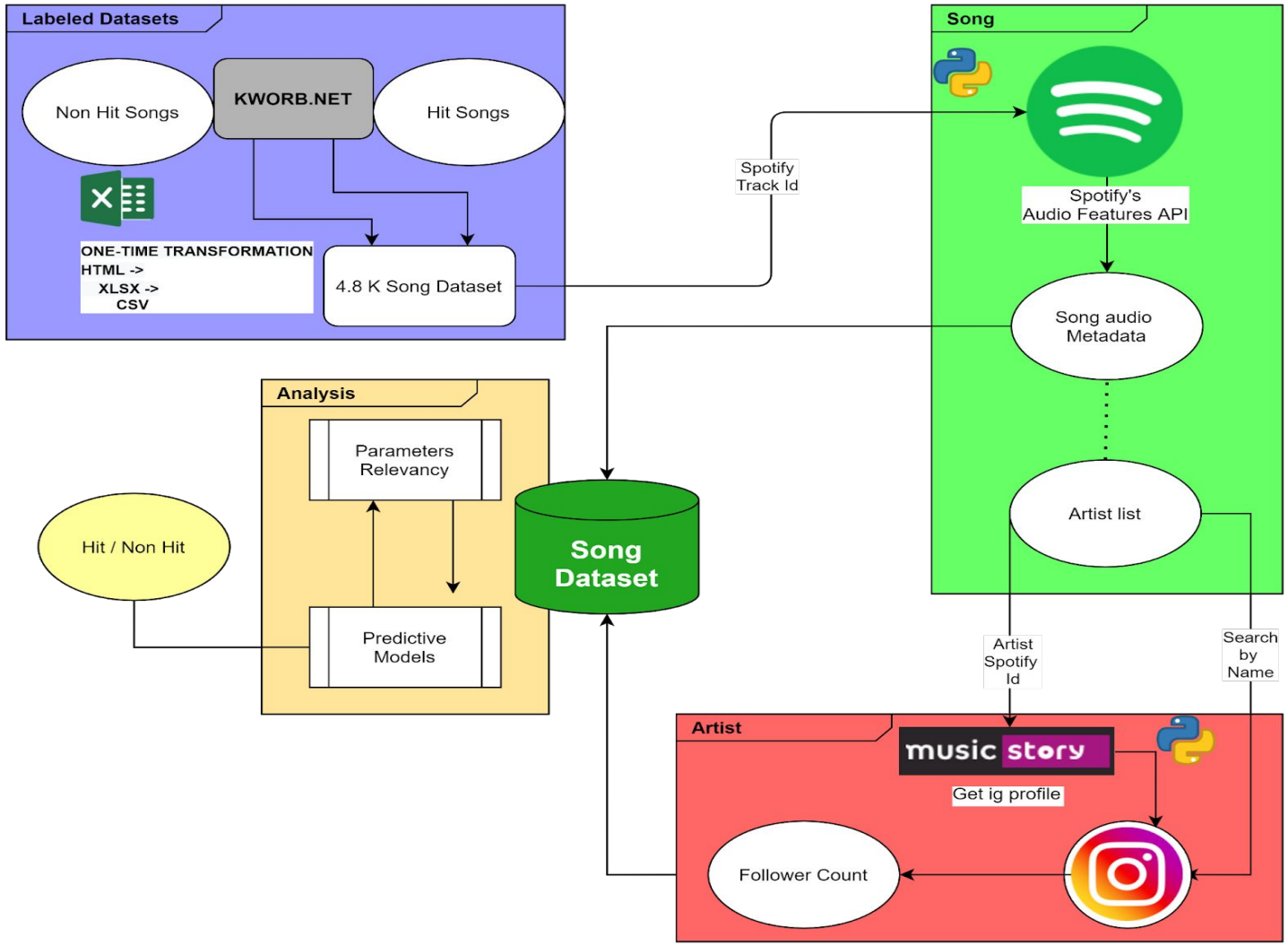
instagram.py × | ig_search_cranberries.json | JsonViewer | decorate_artist_list.py | UNION MSI01-IS01-IS02-IS03-IS04-I

.vscode > instagram.py > { } instaloader

```python
import instaloader
import time

# #of seconds to wait before instagram requests
sleep_time_before_request = 9
# # of records to iterate to find max follower count at instagram search results
max_top_search_results = 8

class InstagramArtist:
    # This is just a POJO to exchange the relevant instagram artist information around our functions
    # or should I say POPO (Plain Old Python Object)
    def __init__(self, data = "", name = "", username = "", followers = -1):
        self.data = data
        self.name = name
        self.username = username
        self.followers = followers

    def get_url(self):
        return "https://www.instagram.com/" + self.username if self.username else ""

class Instagram:
    # This will be the instaloaer wrapper to hide some of its complexity and include some business rules
    def __init__(self, user, password):
        # Keeps user session alive through this object lifecycle
        self.loader = instaloader.Instaloader()
        time.sleep(sleep_time_before_request)
        self.loader.login(user, password)

    def get_artist(self, instgram_handler):
        return_value_artist = InstagramArtist()
        try:
            if instgram_handler:
                time.sleep(sleep_time_before_request)
                profile = instaloader.Profile.from_username(self.loader.context, instgram_handler)
                return_value_artist.data = profile._node
                return_value_artist.followers = profile.followers
        except:
            #silent error, I'm deeply sorry I'm on a rush
```

DATA SOURCES




Labeled Datasets
Non Hit Songs
KWORB.NET
Hit Songs
ONE-TIME TRANSFORMATION
HTML ->
XLSX ->
CSV
4.8 K Song Dataset
Spotify
Track Id
Song
Spotify's
Audio Features API
Song audio
Metadata
Artist list
Analysis
Parameters
Relevancy
Hit / Non Hit
Predictive
Models
Song
Dataset
Artist
Spotify
Id
Search
by
Name
Artist
music story
Get ig profile
Follower Count

VARIABLES DESCRIPTION

	VARIABLE	SOURCE	VARIABLE DESCRIPTION	VARIABLE TYPE
Song Information	Pos	KWORB	Position of the song in the table	Quantitative: Numerical
	artist_and_title	KWORB	Name of the artist and name of the song	Categorical: Nominal
	song_name	KWORB	Name of the song	Categorical: Nominal
	name	SPOTIFY API	Name of the song	Categorical: Nominal
	album	SPOTIFY API	Name of the album	Categorical: Nominal
	weeks	KWORB	How many times the song has been in the list	Quantitative: Numerical
	T10	KWORB	How many weeks the song has been in the top 10	Quantitative: Numerical
	Pk	KWORB	Top position of the song in the table	Quantitative: Numerical
	X	KWORB	How many weeks the song has been in the Pk position: (x3)	Quantitative: Numerical
	pk_streams	KWORB	Highest number of plays of the song in one week	Quantitative: Numerical
	total	KWORB	Total number of plays on spotify	Quantitative: Numerical
	artist_url	KWORB	URL of the artist on Spotify	Categorical: nominal
	track	KWORB	URL of the artist on Spotify	Categorical: Nominal
	release_date	SPOTIFY API	date the song was released	Quantitative: Date
	release_month_date	SPOTIFY API	Month the song was released	Categorical: Ordinal
	Popularity	SPOTIFY API	The popularity of the track. The value will be	Quantitative:
			between 0.0 and 1.0, with 1.0 being the most	numerical
			popular. Popularity is track-based and a measure of how many plays a track received and how recent those plays are. Then artist popularity is derived from that.	

	VARIABLE	SOURCE	VARIABLE DESCRIPTION	VARIABLE TYPE
Audio Features	Acousticness	SPOTIFY API	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	Quantitative: numerical
	Danceability	SPOTIFY API	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.	Quantitative: numerical
	length	SPOTIFY API	The duration of the track in milliseconds.	Quantitative: numerical
	Energy	SPOTIFY API	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.	Quantitative: numerical
	Instrumentalness	SPOTIFY API	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentals value is to 1.0 , the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.	Quantitative: numerical

	VARIABLE	SOURCE	VARIABLE DESCRIPTION	VARIABLE TYPE
Audio Features	Liveness	SPOTIFY API	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.	Quantitative: numerica
	Loudness	SPOTIFY API	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.	Quantitative: numerical
	speechiness	SPOTIFY API	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.666 describe tracks that are probably made entirely of spoken words. Values between 0.333 and 0.666 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.333 most likely represent music and other non-speech-like tracks.	Quantitative: numerical
	tempo	SPOTIFY API	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.	Quantitative: numerical
	time_signature	SPOTIFY API	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).	Quantitative: numerical
	mode	SPOTIFY API	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.	Categorical: binary (Major/minor)
	valence	SPOTIFY API	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).	Quantitative: numerical
	key	SPOTIFY API	The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.	Categorical: Ordinal

	VARIABLE	SOURCE	VARIABLE DESCRIPTION	VARIABLE TYPE
Artist Information	artist_name	KWORB	Name of the artist	Categorical: Nominal
	artist	SPOTIFY API	Name of the artist	Categorical: nominal
	instagram_followers		Number of followers the artist has on instagram	Quantitative: Numerical

VARIABLE	SOURCE	VARIABLE DESCRIPTION	CHALLENGES
Song Genre	Wikipedia, Spotify, Audio Signal processing	Song's genre	Spotify provides the list of genres an artist includes in it's albums. But that doesn't necessarily provide a given song's genre.
			Another option is to analyze by the Spotify audio_analysis API. This is highly complex.
			Another option is to extract it from wikipedia.
Artist age	Wikipedia	Age of the artist.	MusicStory or MusicBrainz opened up this capability for us. Because we could navigate effectively from band to artists and from artists to wikipedia articles.
		Or the array of a band's members age	But, even if we would've been able to do that navigation, wikipedia articles or wikimedia documents do not provide exact metadata. We would've needed to do some text analysis.
Season	Spotify	Season the song was released.	We need to navigate from song to album and find the release date.
		For example Summer or Winter	From there, we could've set Northern / Southern Hemisphere seasons.
			But, we wanted also to have the artists/band main country, which takes us also to the navigate/scrap wikipedia challenge.
Country	Wikipedia	Country of the artist	Navigate/scrap wikipedia challenge.
Key words	Lyrics provider	The frequency of the words from the song lyrics.	We could go from song to lyrics using lyrics provider. Some of these are not free. The ones that are popular and have an API available are:
			https://www.musixmatch.com/
			https://www.lyrics.com/
Chorus frequency	Lyrics provider	number of times the chorus of the song is repeated	Hag
Lyrics Sentiment	Lyrics provider	The sentiment analysis from the song lyrics	Having the lyrics we could've analyze:
			Polarity from the song title nouns. Meaning if it is Negative, Positive or Neutral to that noun.
			Emotional states: anger, disgust, joy, fear.
language of the song	Wikipedia / Lyrics provider	In which language(s) the song is.	We don't know whether it would've been easier to do lyrics analysis or the wikipedia scrapping. Probably the latter.
chord progression	song sheet music	What chord progressions does the song have	Time and we need more people who know how to read music. It is difficult to convert data into numbers
Featuring artists	Spotify	if the song is sung by one or more artists (one or more musical groups)	We needed to spend more time with the Spotify APIs, and we decided this piece of information was not amongst our top priorities.
Band	Spotify	Whether the song is performed by a band or a solo artist.	We needed to spend more time with the Spotify APIs, and we decided this piece of information was not amongst our top priorities.

UNIVARIATE ANALYSIS

DESCRIBE THE SAMPLE WITH DESCRIPTIVE STATISTICS

- CATEGORICAL: frequency table
- QUANTITATIVE: descriptive analysis
  - measures of central tendency
  - measures of dispersion

"release_month" on Sample - (12 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	0	0.0 %
Invalid	0	0.0 %
Empty	0	0.0 %

0 UNIQUES 0.0 %

0 INVALIDS 0.0 %

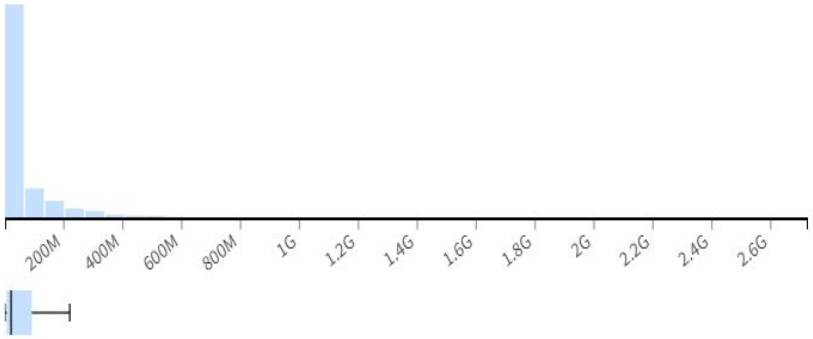
Top 12 out of 12 values in sample	Count	%	Cum. %
1	458	10.7	10.7
11	456	10.6	21.3
3	449	10.5	31.8
5	404	9.4	41.2
10	379	8.8	50.1
8	349	8.1	58.2
6	339	7.9	66.1
4	327	7.6	73.8
9	300	7.0	80.8
12	294	6.9	87.6
2	292	6.8	94.4
7	[illegible]	[illegible]	[illegible]

"Total" on Sample - (3772 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	3,378	78.8 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	166910
Max	2726385847
Mean	97492619
Median	20693870
StdDev	205902718
Mode	5327027
Distinct	3772
IQR	85653262
Sum	417853363631

TOP VALUES

164355425	5	0.1 %
228123561	5	0.1 %
30373370	5	0.1 %
5327027	5	0.1 %
5560683	5	0.1 %
7442276	5	0.1 %
107267766	4	0.1 %

More values and actions

OUTLIERS (0 low, 521 high)

221217583
221286400
221455894
221632774
222884921
225019037

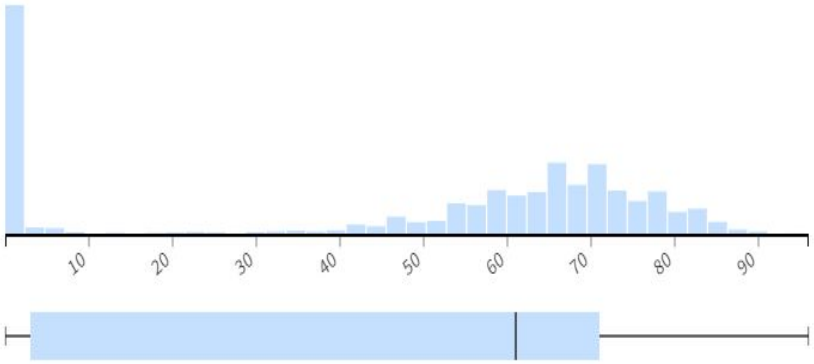
TOP INVALIDS

"popularity" on Sample - (96 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	1	0.0 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0
Max	96
Mean	47.582
Median	61
StdDev	30.661
Mode	0
Distinct	96
IQR	68
Sum	203936

TOP VALUES

0	928	21.7 %
68	120	2.8 %
65	117	2.7 %
71	115	2.7 %
67	113	2.6 %
69	111	2.6 %
73	108	2.5 %

More values and actions

OUTLIERS (0 low, 0 high)

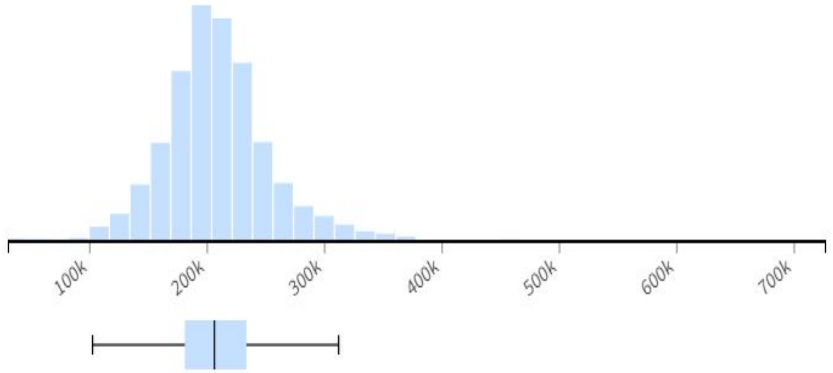
TOP INVALIDS

"length" on Sample - (3468 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	2,895	67.5 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	30133
Max	727106
Mean	210146
Median	206099
StdDev	50410
Mode	142417
Distinct	3468
IQR	52403
Sum	900687408

TOP VALUES

142417	6	0.1 %
169038	6	0.1 %
173653	6	0.1 %
209453	6	0.1 %
211223	6	0.1 %
222044	6	0.1 %
278719	6	0.1 %

More values and actions

OUTLIERS (39 low, 148 high)

30133
30582
35240
37013
37640
38333

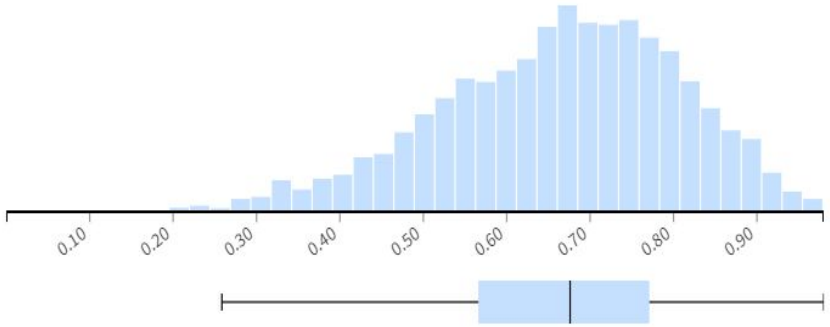
TOP INVALIDS

"danceability" on Sample - (666 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	99	2.3 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0
Max	0.98000
Mean	0.66339
Median	0.67600
StdDev	0.14552
Mode	0.74600
Distinct	666
IQR	0.20500
Sum	2843.3

TOP VALUES

0.746	23	0.5 %
0.681	22	0.5 %
0.695	21	0.5 %
0.652	20	0.5 %
0.664	20	0.5 %
0.719	20	0.5 %
0.534	19	0.4 %

More values and actions

OUTLIERS (23 low, 0 high)

0
0.13000
0.13400
0.15000
0.15300
0.18400

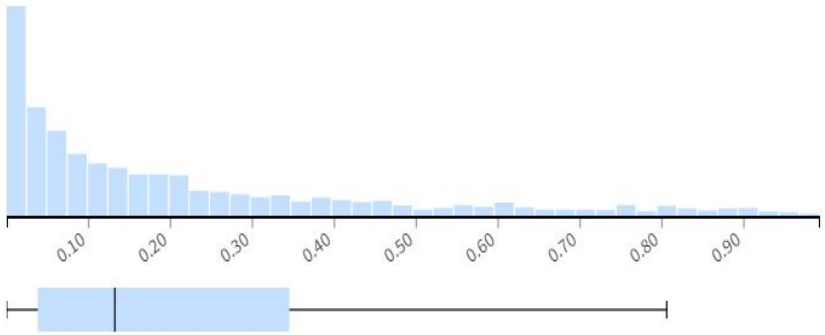
TOP INVALIDS

"acousticness" on Sample - (1681 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	711	16.6 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0.000011500
Max	0.99300
Mean	0.23060
Median	0.13200
StdDev	0.25099
Mode	0.11400
Distinct	1681
IQR	0.30720
Sum	988.34

TOP VALUES

0.114	19	0.4 %
0.139	19	0.4 %
0.128	18	0.4 %
0.166	17	0.4 %
0.111	16	0.4 %
0.112	16	0.4 %
0.125	16	0.4 %

More values and actions

OUTLIERS (0 low, 212 high)

0.80600
0.80700
0.80800
0.80900
0.81000
0.81100

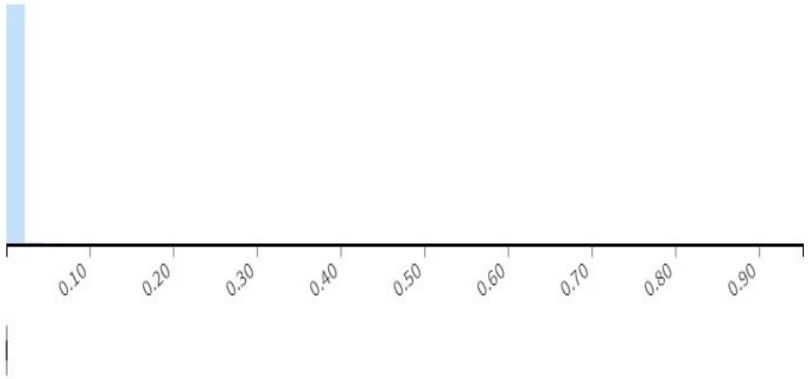
TOP INVALIDS

"instrumentalness" on Sample - (1268 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	859	20.0 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0
Max	0.95300
Mean	0.013366
Median	0
StdDev	0.077757
Mode	0
Distinct	1268
IQR	0.000061300
Sum	57.285

TOP VALUES

0.0	2310	53.9 %
1.13E-4	7	0.2 %
1.65E-6	7	0.2 %
8.75E-5	7	0.2 %
0.00329	6	0.1 %
1.18E-5	6	0.1 %
1.32E-4	6	0.1 %

More values and actions

OUTLIERS (0 low, 869 high)

0.00015500
0.00015600
0.00015700
0.00015900
0.00016000
0.00016100

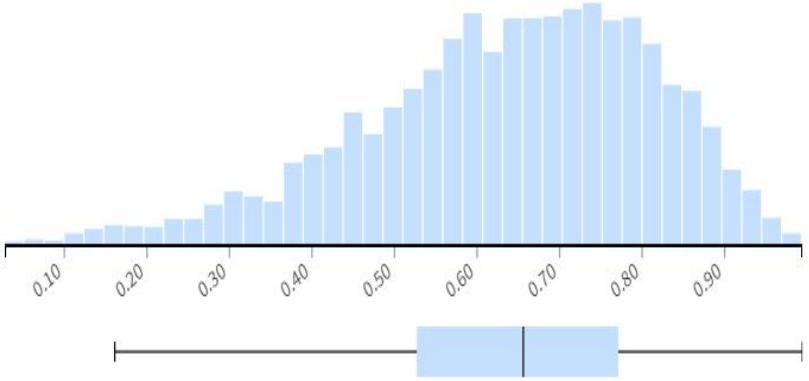
TOP INVALIDS

"energy" on Sample - (752 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	113	2.6 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0.027900
Max	0.99400
Mean	0.63567
Median	0.65600
StdDev	0.17748
Mode	0.70300
Distinct	752
IQR	0.24400
Sum	2724.5

TOP VALUES

0.703	21	0.5 %
0.741	21	0.5 %
0.71	19	0.4 %
0.773	18	0.4 %
0.567	17	0.4 %
0.595	17	0.4 %
0.693	17	0.4 %

More values and actions

OUTLIERS (59 low, 0 high)

0.027900
0.031600
0.054000
0.056100
0.056500
0.066800

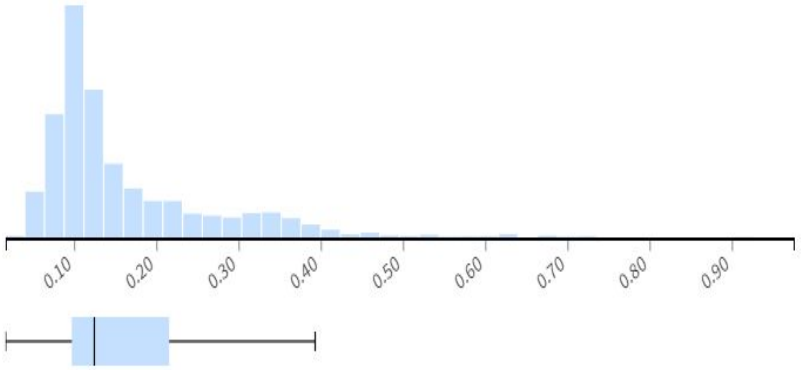
TOP INVALIDS

"liveness" on Sample - (912 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	269	6.3 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0.016200
Max	0.97600
Mean	0.17914
Median	0.12400
StdDev	0.13855
Mode	0.10400
Distinct	912
IQR	0.11840
Sum	767.78

TOP VALUES

0.104	66	1.5 %
0.113	55	1.3 %
0.106	54	1.3 %
0.111	53	1.2 %
0.108	51	1.2 %
0.11	45	1.0 %
0.107	44	1.0 %

More values and actions

OUTLIERS (0 low, 291 high)

0.39300
0.39400
0.39500
0.39600
0.39700
0.39900

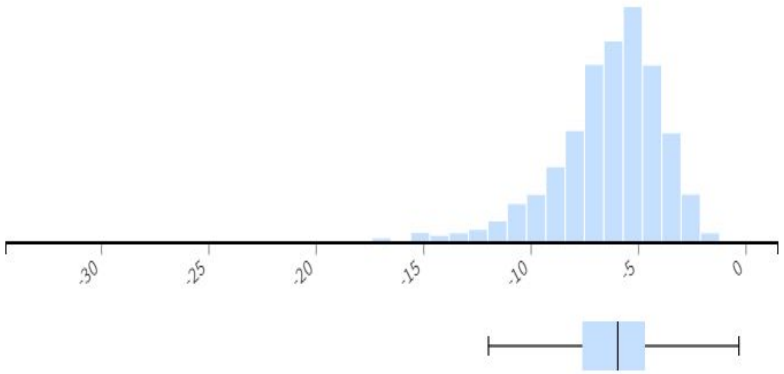
TOP INVALIDS

"loudness" on Sample - (2955 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	2,066	48.2 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	-34.475
Max	1.5090
Mean	-6.4490
Median	-5.9660
StdDev	2.6894
Mode	-6.7610
Distinct	2955
IQR	2.9140
Sum	-27640

TOP VALUES

-4.835	7	0.2 %
-5.523	7	0.2 %
-6.761	7	0.2 %
-12.472	6	0.1 %
-3.537	6	0.1 %
-4.228	6	0.1 %
-4.31	6	0.1 %

More values and actions

OUTLIERS (164 low, 2 high)

-34.475
-25.166
-23.447
-23.023
-22.282
-21.992

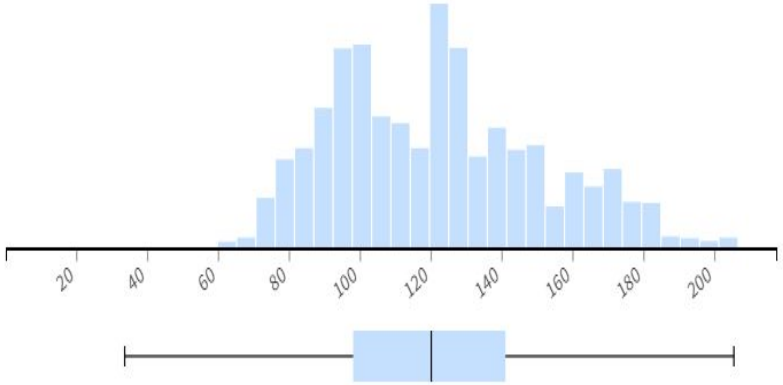
TOP INVALIDS

"tempo" on Sample - (3363 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	2,727	63.6 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0
Max	217.65
Mean	121.79
Median	120.02
StdDev	29.646
Mode	108.00
Distinct	3363
IQR	42.967
Sum	521987

TOP VALUES

108.003	8	0.2 %
100.021	7	0.2 %
122.061	6	0.1 %
138.03	6	0.1 %
139.413	6	0.1 %
139.631	6	0.1 %
75.014	6	0.1 %

More values and actions

OUTLIERS (1 low, 7 high)

0
205.90
205.94
211.84
212.12
217.65

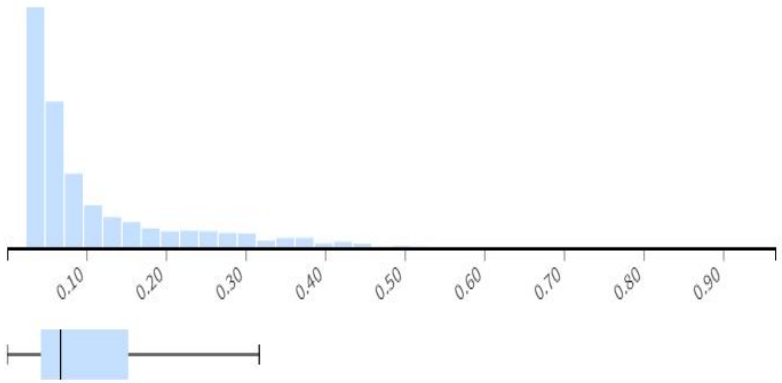
TOP INVALIDS

"speechiness" on Sample - (1061 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	210	4.9 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0
Max	0.96600
Mean	0.11856
Median	0.067000
StdDev	0.11652
Mode	0.045000
Distinct	1061
IQR	0.10970
Sum	508.15

TOP VALUES

0.045	18	0.4 %
0.105	18	0.4 %
0.0479	17	0.4 %
0.109	16	0.4 %
0.15	16	0.4 %
0.0363	15	0.3 %
0.102	15	0.3 %

More values and actions

OUTLIERS (0 low, 345 high)

0.31700
0.31800
0.31900
0.32000
0.32300
0.32400

TOP INVALIDS

"time_signature" on Sample - (5 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	1	0.0 %
Invalid	0	0.0 %
Empty	0	0.0 %

1 UNIQUES 0.0 %

- 0

0 INVALIDS 0.0 %

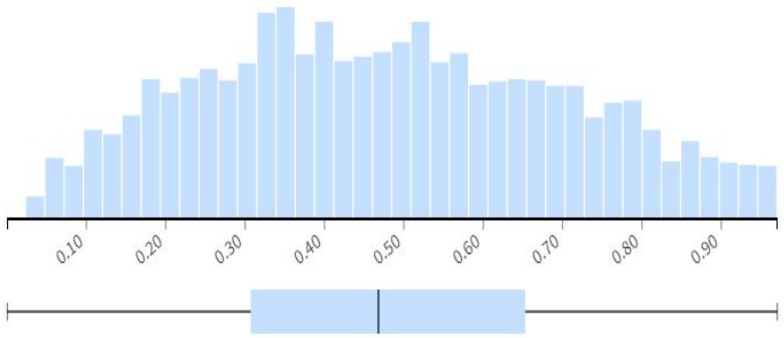
Top 5 out of 5 values in sample	Count	%	Cum. %
4	4038	94.2	94.2
3	169	3.9	98.2
5	66	1.5	99.7
1	12	0.3	100.0
0	1	0.0	100.0

"valence" on Sample - (929 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	157	3.7 %
Invalid	0	0.0 %
Empty	0	0.0 %



STATISTICS

Min	0
Max	0.97100
Mean	0.47913
Median	0.46800
StdDev	0.22598
Mode	0.46000
Distinct	929
IQR	0.34600
Sum	2053.5

TOP VALUES

0.46	19	0.4 %
0.528	18	0.4 %
0.405	17	0.4 %
0.468	17	0.4 %
0.35	15	0.3 %
0.351	15	0.3 %
0.17	14	0.3 %

More values and actions

OUTLIERS (0 low, 0 high)

TOP INVALIDS

"mode" on Sample - (2 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	0	0.0 %
Invalid	0	0.0 %
Empty	0	0.0 %

0 UNIQUES 0.0 %

0 INVALIDS 0.0 %

Top 2 out of 2 values in sample	Count	%	Cum. %
1	2483	57.9	57.9
0	1803	42.1	100.0

"key" on Sample - (12 distinct)

CATEGORICAL | NUMERICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	0	0.0 %
Invalid	0	0.0 %
Empty	0	0.0 %

0 UNIQUES 0.0 %

0 INVALIDS 0.0 %

Top 12 out of 12 values in sample	Count	%	Cum. %
1	605	14.1	14.1
0	455	10.6	24.7
7	402	9.4	34.1
11	396	9.2	43.4
9	377	8.8	52.1
5	352	8.2	60.4
2	335	7.8	68.2
6	332	7.7	75.9
8	324	7.6	83.5
10	308	7.2	90.7
4	277	6.5	97.1
3	[illegible]	[illegible]	[illegible]

"artist" on Sample - (914 distinct)

CATEGORICAL | VALUES CLUSTERING

SUMMARY

Valid	4,286	100.0 %
Unique	444	10.4 %
Invalid	0	0.0 %
Empty	0	0.0 %

444 UNIQUES 10.4 %

- *NSYNC
- 13 Organisé
- 187 Strassenbande
- 347aidan

0 INVALIDS 0.0 %

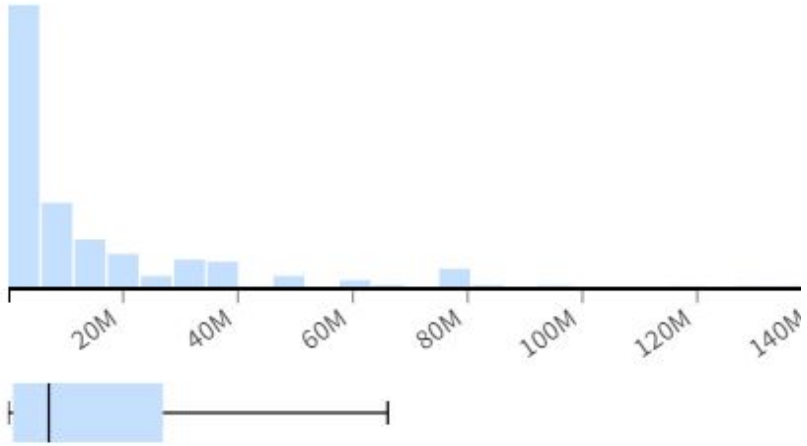
Top 50 out of 914 values in sample

- Drake
- Ariana Grande
- BTS
- The Weeknd
- Various Artists
- Taylor Swift
- Bad Bunny
- Kanye West
- Eminem
- One Direction
- Justin Bieber
- Ed Sheeran

agram_followers" on Sample - (947 distinct)

NUMERICAL | VALUES CLUSTERING

4,286	100.0 %
463	10.8 %
0	0.0 %
0	0.0 %



949578
79227
7897
27232
42326
08563
632565223

TOP VALUES		
7.7942326E7	138	3.2 %
2.30949578E8	71	1.7 %
3.9784356E7	69	1.6 %
3.0162766E7	65	1.5 %
1.53355792E8	61	1.4 %
3.2104493E7	54	1.3 %
980899.0	53	1.2 %

More values and actions

OUTLIERS (0 low, 487

70333325
77942326
81158583
88637064
93996380
102471163

BIVARIATE ANALYSIS

DESCRIBE THE RELATIONSHIP BETWEEN VARIABLES.

Spearman's correlation:

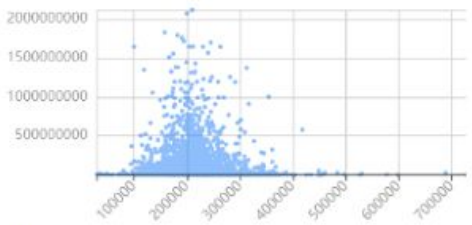
- evaluates the monotonic relationship between two continuous or ordinal variables.
- monotonic relationship: the variables tend to change at the same time, but not necessarily at a constant rate.

Correlation matrix on 21 variables (Spearman) No split

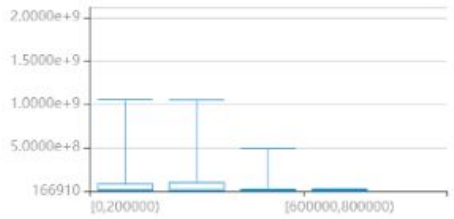
	Total	Wks	T10	Pk	PkStreams	length	popularity	danceability	acousticness	energy	instrumen...	liveness	loudness	speechin
Total	1.000	0.919	0.635	-0.799	0.724	-0.020	0.268	0.127	0.001	-0.005	-0.067	-0.043	0.081	0.
Wks	0.919	1.000	0.666	-0.689	0.440	0.056	0.152	0.078	-0.044	0.039	-0.058	-0.048	0.101	-0.
T10	0.635	0.666	1.000	-0.643	0.108	0.091	-0.021	0.069	-0.034	0.079	0.006	-0.104	0.107	-0.
Pk	-0.799	-0.689	-0.643	1.000	-0.765	-0.030	-0.189	-0.050	-0.004	0.049	0.043	0.009	-0.001	-0.
PkStreams	0.724	0.440	0.108	-0.765	1.000	-0.121	0.350	0.124	0.079	-0.091	-0.055	-0.010	-0.006	0.
length	-0.020	0.056	0.091	-0.030	-0.121	1.000	-0.075	-0.175	-0.039	0.043	0.059	-0.006	0.002	-0.
popularity	0.268	0.152	-0.021	-0.189	0.350	-0.075	1.000	0.102	0.045	-0.029	-0.026	-0.046	0.042	0.
danceability	0.127	0.078	0.069	-0.050	0.124	-0.175	0.102	1.000	-0.111	0.016	-0.018	-0.085	0.100	0.
acousticness	0.001	-0.044	-0.034	-0.004	0.079	-0.039	0.045	-0.111	1.000	-0.443	-0.007	-0.077	-0.317	-0.
energy	-0.005	0.039	0.079	0.049	-0.091	0.043	-0.029	0.016	-0.443	1.000	-0.071	0.099	0.727	0.
instrument...	-0.067	-0.058	0.006	0.043	-0.055	0.059	-0.026	-0.018	-0.007	-0.071	1.000	-0.056	-0.215	-0.
liveness	-0.043	-0.048	-0.104	0.009	-0.010	-0.006	-0.046	-0.085	-0.077	0.099	-0.056	1.000	0.058	0.
loudness	0.081	0.101	0.107	-0.001	-0.006	0.002	0.042	0.100	-0.317	0.727	-0.215	0.058	1.000	0.
speechiness	0.003	-0.053	-0.069	-0.008	0.075	-0.099	0.038	0.319	-0.054	0.085	-0.148	0.064	0.036	1.
tempo	-0.013	-0.018	-0.096	0.010	0.008	0.008	0.028	-0.029	-0.084	0.112	-0.006	0.004	0.077	0.
time_signat...	0.026	0.025	0.072	-0.013	0.015	-0.004	0.031	0.126	-0.105	0.134	-0.007	-0.005	0.099	0.
mode	0.016	0.025	-0.030	-0.016	-0.001	0.040	0.016	-0.097	0.032	-0.025	-0.060	-0.024	-0.015	-0.
valence	0.075	0.075	0.150	0.018	0.026	-0.162	0.046	0.326	-0.055	0.381	-0.141	-0.024	0.340	0.
key	-0.007	-0.004	0.038	0.032	-0.017	-0.035	0.000	0.017	0.005	0.015	0.001	0.002	0.003	0.
release_mo...	0.035	0.037	0.032	-0.005	0.010	0.019	0.041	-0.061	0.026	-0.009	-0.018	0.017	-0.009	-0.
instagram_f...	0.089	0.016	-0.067	-0.239	0.221	0.050	0.067	-0.009	-0.038	-0.041	-0.039	-0.004	0.025	0.

Total by # length

Scatter plot



Box Plot



Histogram

Mosaic plot

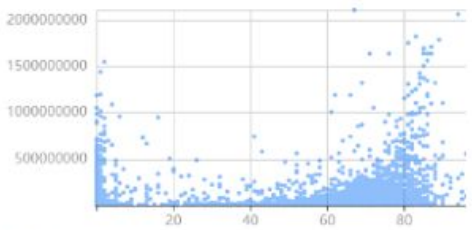
Summary

Spearman	-0.019528891
p-value	0.2012112247
Pearson	-0.010732462
p-value	0.4824538795
N values	4286
Covariance	-1.09282e+11
Kendall tau	-0.011827457
p-value	0.2457455012
N finite	4285

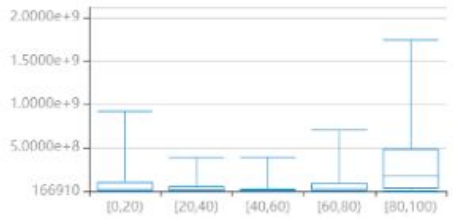
Frequency table

Total by # popularity

Scatter plot



Box Plot



Histogram

Mosaic plot

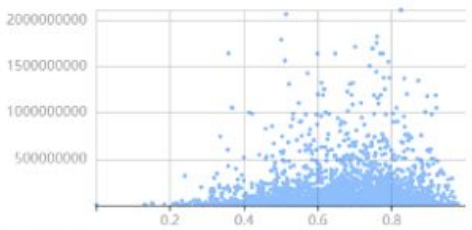
Summary

Spearman	0.2677671313
p-value	2.899668e-71
Pearson	0.1163679968
p-value	2.152264e-14
N values	4286
Covariance	720573509.22
Kendall tau	0.1960862591
p-value	1.743952e-78
N finite	4285

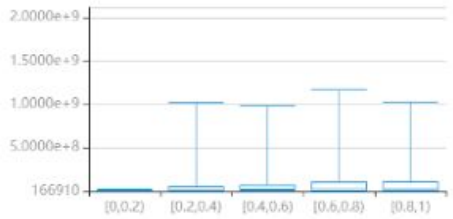
Frequency table

Total by # danceability

Scatter plot



Box Plot



Histogram

Mosaic plot

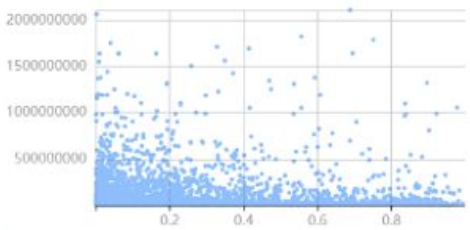
Summary

Spearman	0.1267854536
p-value	8.044743e-17
Pearson	0.068474022
p-value	0.0000072484
N values	4286
Covariance	2012477.229
Kendall tau	0.0846077878
p-value	1.080749e-16
N finite	4285

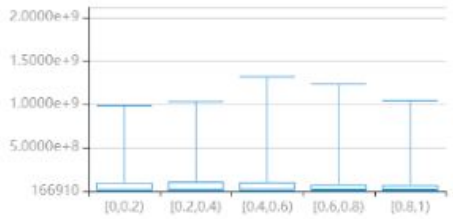
Frequency table

Total by # acousticness

Scatter plot



Box Plot



Histogram

Mosaic plot

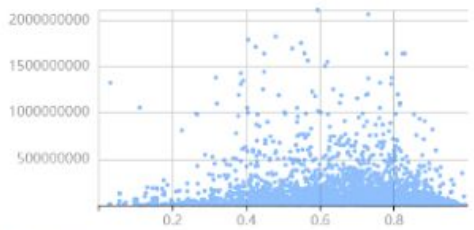
Summary

Spearman	0.0010911415
p-value	0.9430752634
Pearson	-0.003936526
p-value	0.7967075651
N values	4286
Covariance	-199531.6716
Kendall tau	0.0006814635
p-value	0.9466965759
N finite	4285

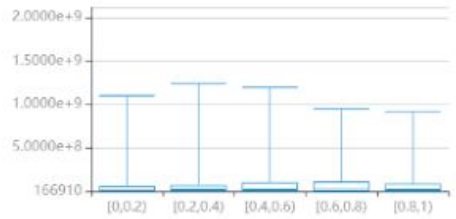
Frequency table

Total by # energy

Scatter plot



Box Plot



Histogram

Mosaic plot

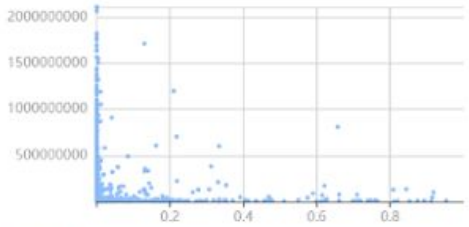
Summary

Spearman	-0.004925102
p-value	0.7472219026
Pearson	-0.015234637
p-value	0.3187524218
N values	4286
Covariance	-546175.4695
Kendall tau	-0.002839996
p-value	0.7806354265
N finite	4285

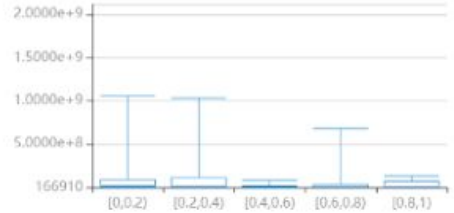
Frequency table

Total by # instrumentalness

Scatter plot



Box Plot



Histogram

Mosaic plot

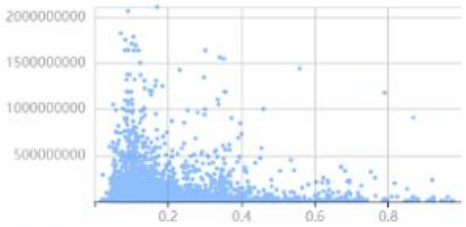
Summary

Spearman	-0.067498218
p-value	0.0000097709
Pearson	-0.028940269
p-value	0.0581889842
N values	4286
Covariance	-454549.3953
Kendall tau	-0.049213041
p-value	0.0000094208
N finite	4285

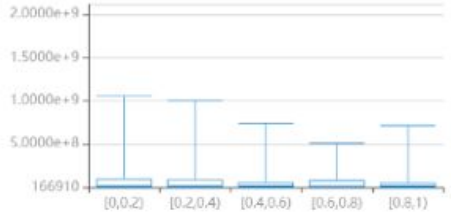
Frequency table

Total by # liveness

Scatter plot



Box Plot



Histogram

Mosaic plot

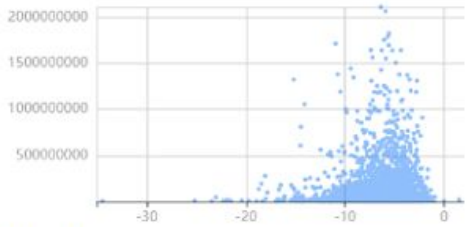
Summary

Spearman	-0.042947916
p-value	0.0049258402
Pearson	-0.033910585
p-value	0.0264340677
N values	4286
Covariance	-948957.8629
Kendall tau	-0.028691083
p-value	0.0049338325
N finite	4285

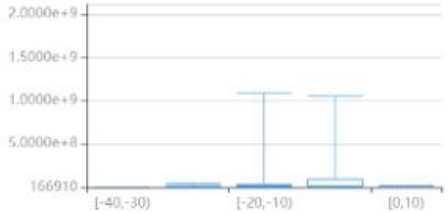
Frequency table

Total by # loudness

Scatter plot



Box Plot



Histogram

Mosaic plot

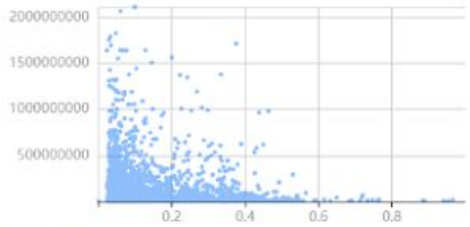
Summary

Spearman	0.0814418514
p-value	9.3723568e-8
Pearson	0.0551278891
p-value	0.000305773
N values	4286
Covariance	29943247.7
Kendall tau	0.0546137123
p-value	8.3433679e-8
N finite	4285

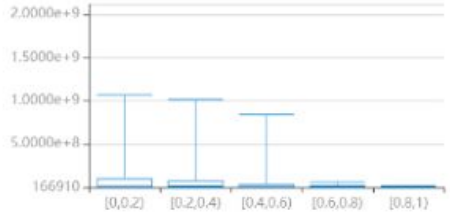
Frequency table

Total by # speechiness

Scatter plot



Box Plot



Histogram

Mosaic plot

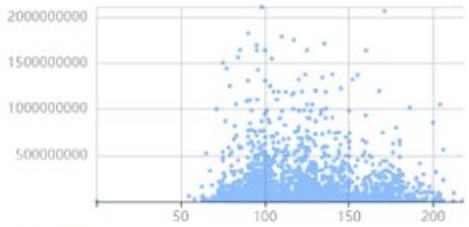
Summary

Spearman	0.0032611322
p-value	0.8310058455
Pearson	-0.050924809
p-value	0.000853839
N values	4286
Covariance	-1198604.431
Kendall tau	0.0015499546
p-value	0.8791637087
N finite	4285

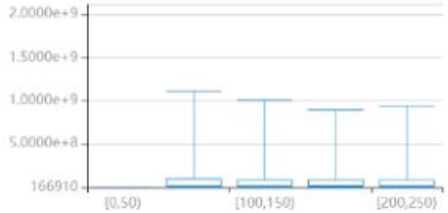
Frequency table

Total by # tempo

Scatter plot



Box Plot



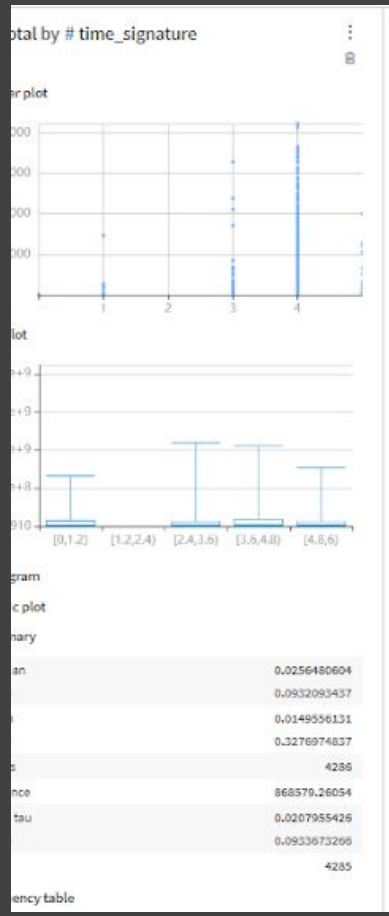
Histogram

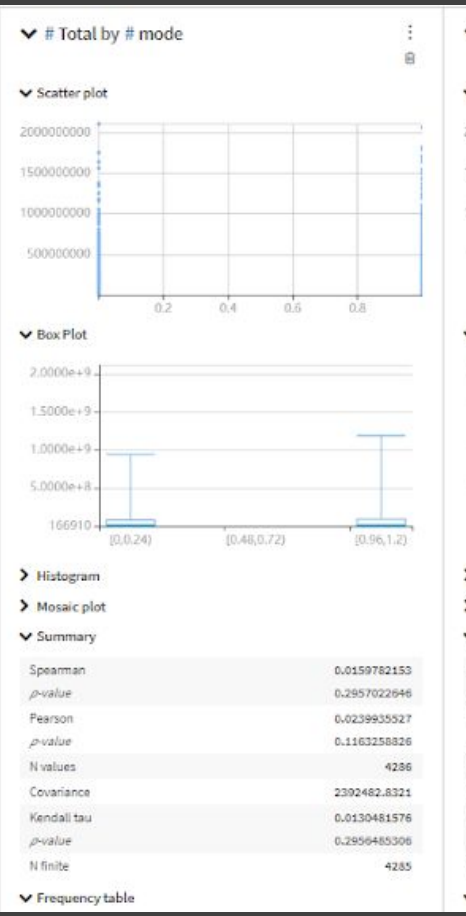
Mosaic plot

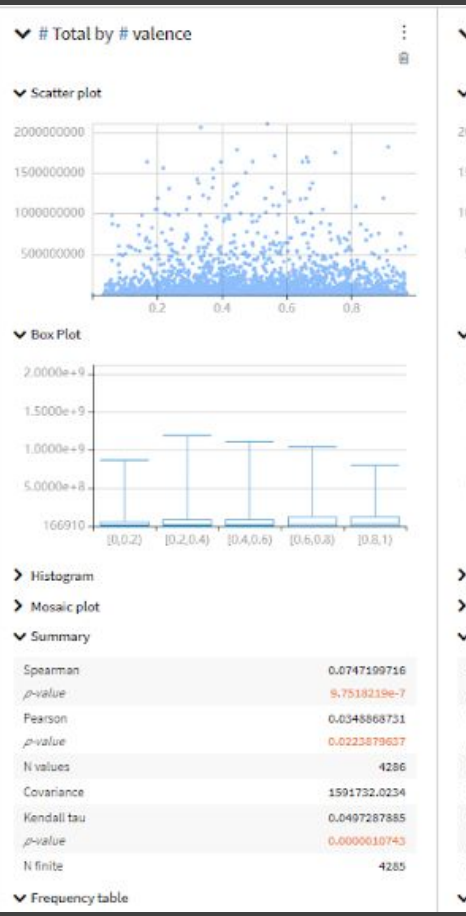
Summary

Spearman	-0.013032899
p-value	0.3937038048
Pearson	-0.018718338
p-value	0.220556157
N values	4286
Covariance	-112082269.8
Kendall tau	-0.008370842
p-value	0.4113585042
N finite	4285

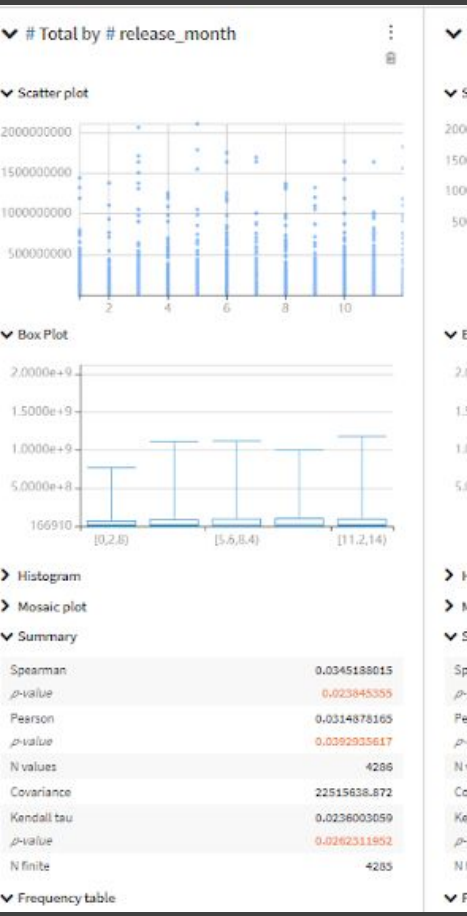
Frequency table

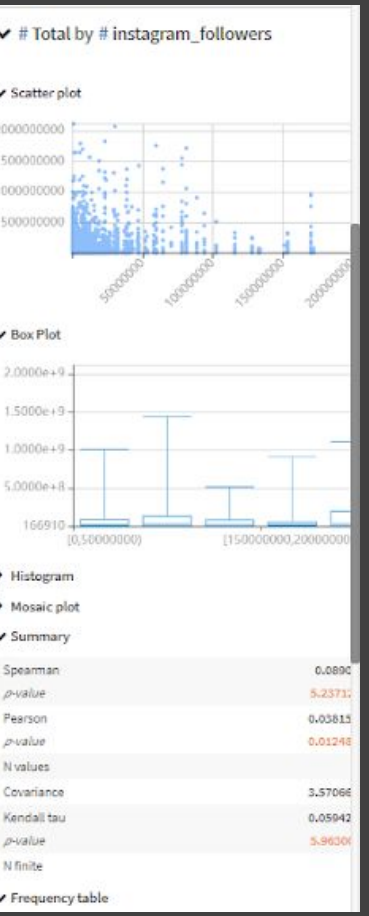

otal by # time_signature
r plot
lot
gram
c plot
nary
an 0.0256480604
0.0932093437
0.0149556131
0.3276974837
4286
nce 868579.26054
tau 0.0207955426
0.0933673266
4285
ency table


Total by # mode
Scatter plot
Box Plot
Histogram
Mosaic plot
Summary
Spearman 0.0159782153
p-value 0.2957022646
Pearson 0.0239935527
p-value 0.1163258826
N values 4286
Covariance 2392482.8321
Kendall tau 0.0130481576
p-value 0.2956485306
N finite 4285
Frequency table


Total by # valence
Scatter plot
Box Plot
Histogram
Mosaic plot
Summary
Spearman 0.0747199716
p-value 9.7518219e-7
Pearson 0.0348868731
p-value 0.0223879637
N values 4286
Covariance 1591732.0234
Kendall tau 0.0497287885
p-value 0.0000010743
N finite 4285
Frequency table

Total by # key
Scatter plot
Box Plot
Histogram
Mosaic plot
Summary
Spearman -0.007491104
p-value 0.62397057
Pearson -0.000282326
p-value 0.9852594314
N values 4286
Covariance -208476.6947
Kendall tau -0.005244122
p-value 0.6220265758
N finite 4285
Frequency table


Total by # release_month
Scatter plot
Box Plot
Histogram
Mosaic plot
Summary
Spearman 0.0345188015
p-value 0.023845355
Pearson 0.0314878165
p-value 0.0392935617
N values 4286
Covariance 22515638.872
Kendall tau 0.0236003059
p-value 0.0262311952
N finite 4285
Frequency table


Total by # instagram_followers
Scatter plot
Box Plot
Histogram
Mosaic plot
Summary
Spearman
p-value
Pearson
p-value
N values
Covariance
Kendall tau
p-value
N finite
Frequency table

Data Preparation

Data Preparation

- Have more than 20% Of Nulls
- Have more than 20% Of Invalid Values
- Dependent variables of streams
- Duplicated values
- Heteroscedasticity

Data Preparation

Special Characters

- Convert Date (Released data)
- Names of the artists (Misspelled)

Elimination of variables

- T10 90%empty values
- X: 96% empty values

Duplicated values

- Artists names
- Songs names
- Artists and title

Data Preparation

- Variables related to the numbers of streams:
  - Pk streams, Weeks, Pk, Popularity
- Heterocedasticity:
  - Release_date with Month_release
- Elimination of rows:
  - Tempo (between 60 and 220)
  - Loudness (between -20 and 0)

After cleaning: 4.273 rows

Experts Insights

Specialists Consultancy

We asked two music experts:

- What features they believe a hit song might have?

- What social characteristics should we take into account?

Specialists Consultancy

First expert insights (Argentinian music teacher):

•Chord progressions that are generally kind to the human ear

- Ex: 90% of hits are made in C tone

•Music genres

•BPM which determines the rhythm

- Ex: The Beatles music are composed in four quarters (easier to be comprehended by people)
-

•Kind melodies / easy to “digest”

•Easy lirycs

Specialists Consultancy

Second expert insights (American composer):

- When he composes, he thinks about danceability
- Tempo (BPM), chord and language are very important
- Major or minor keys
- Message musician wants to transmit (trend words and expressions)
- Music genre is very important
- Music length (90s 3min30sec / Nowadays less than 3min)

Specialists Consultancy

- Season is an important factor
    - Ex: Oct, Nov, Dec slow songs and dance song release Jan (takes time to get into a Spotify playlist and people to listen)
- Usually use chorus 2 or 3 times
- How long the song has been in the top 10
- Artists age as a big factor: 16 to 23

Output Analysis

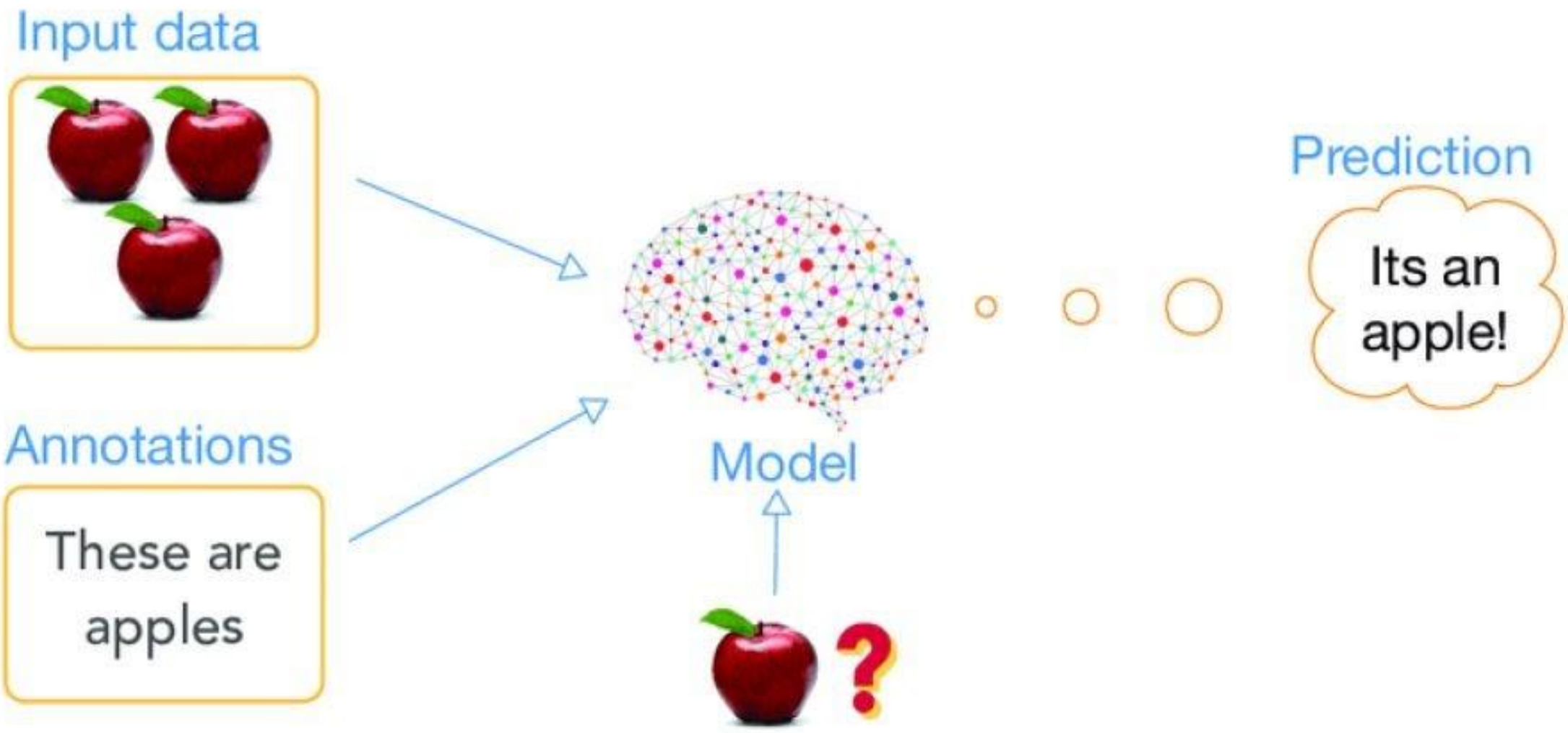
Variables considered into the model

- Length
- Danceability
- Acousticness
- Loudness
- Liveness
- Instrumentalness
- Speechiness
- Tempo
- Time Signature
- Mode
- Valence
- Key
- Released Month
- Instagram followers

Modeling

What are the chosen models? Why? What the output might be.

supervised learning



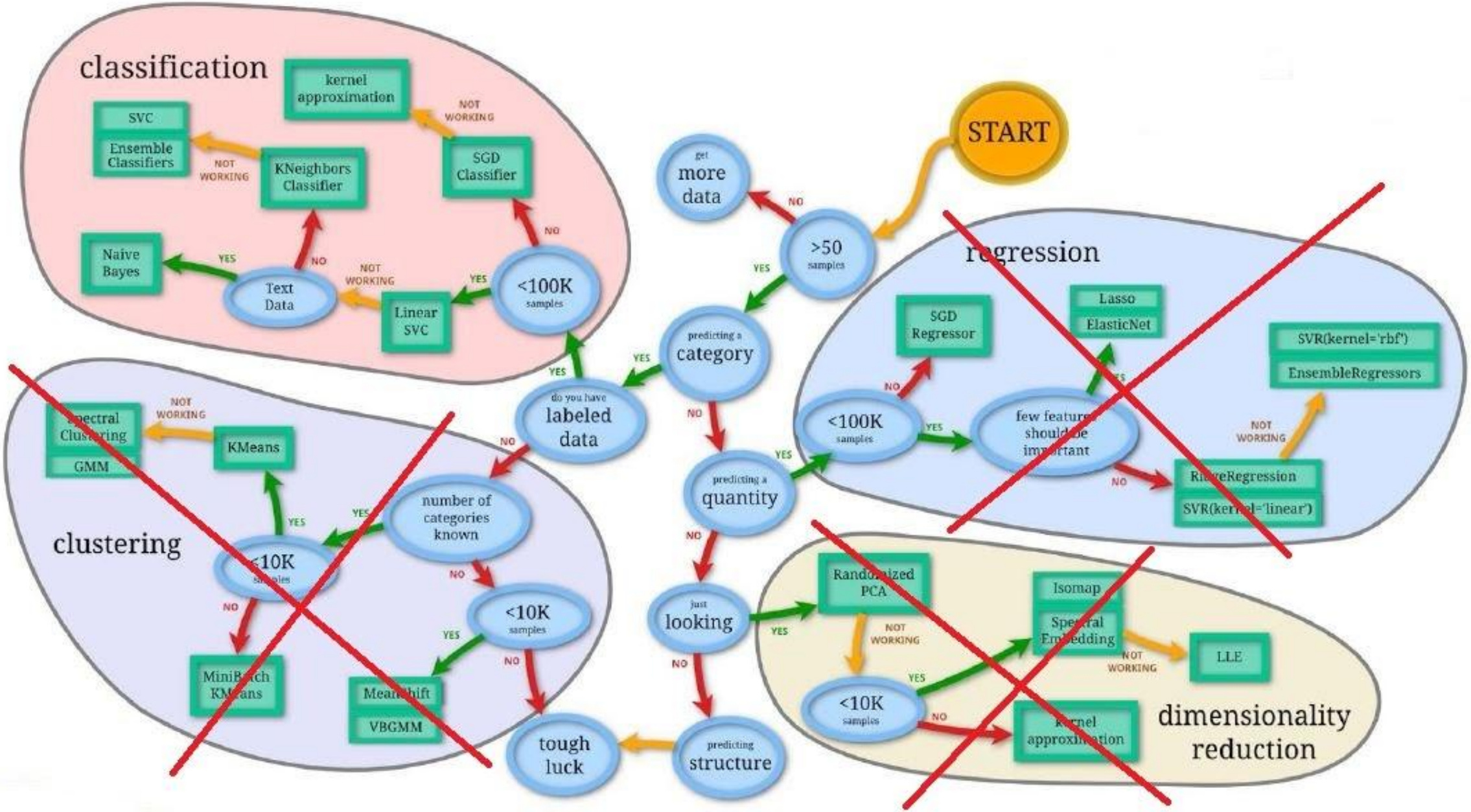
Outputs configuration

- It is very difficult to define if a music is a Hit or not.

- Generation of different outputs columns :

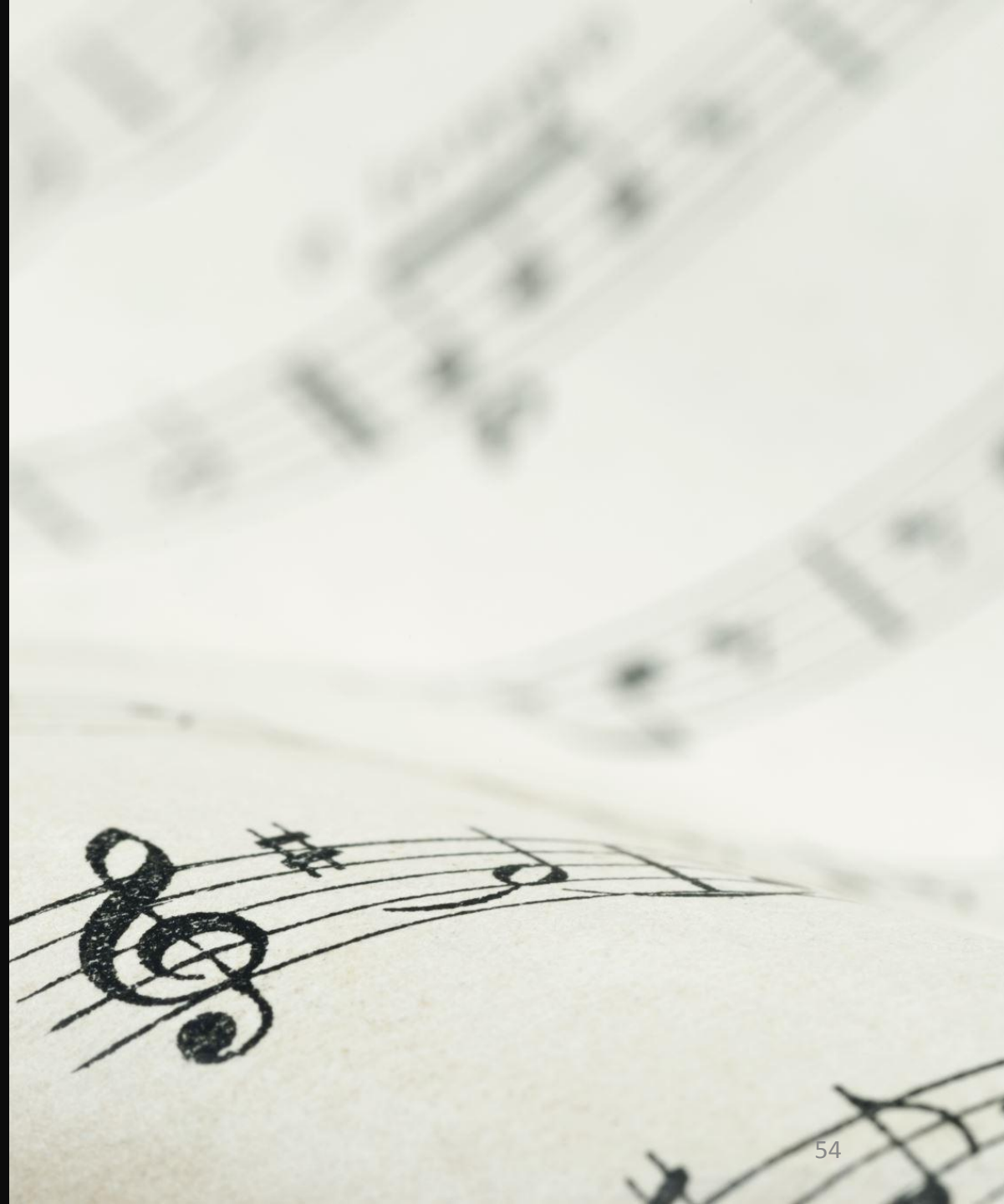
  Top 100 / Top 500 / Top 1000 / Top 2000 / Top 2500

- Finding a balance between the performance of our models and the demands of our predictions

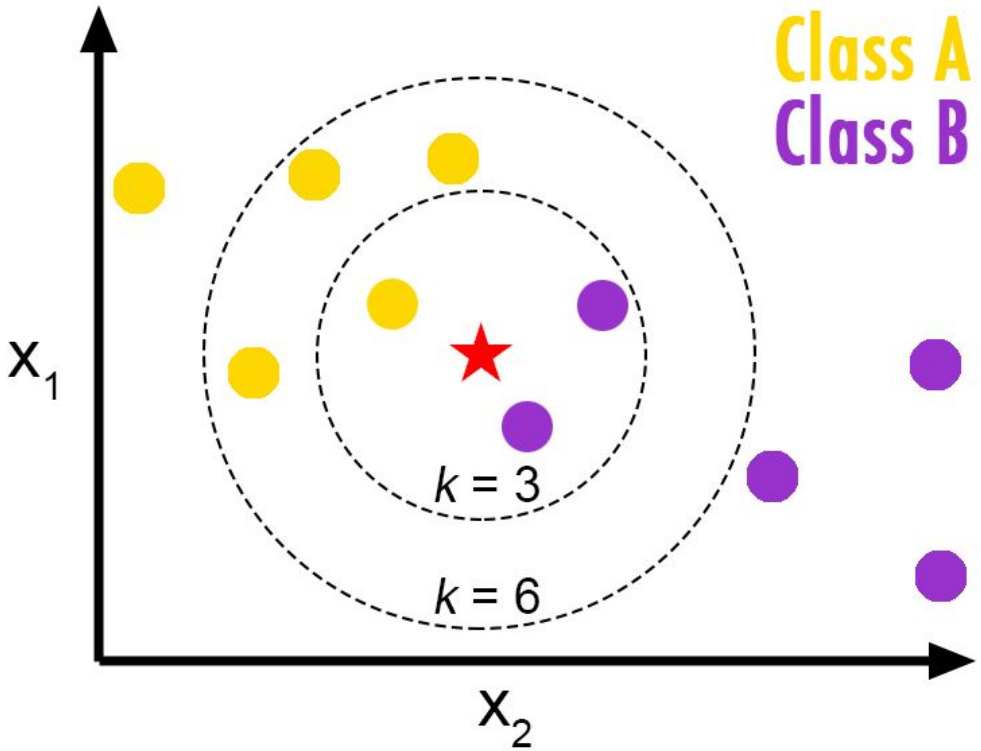

START
>50 samples
NO
get more data
YES
predicting a category
YES
do you have labeled data
NO
predicting a quantity
NO
just looking
NO
predicting structure
tough luck
YES
<100K samples
NO
Linear SVC
NOT WORKING
Text Data
YES
Naive Bayes
NO
KNeighbors Classifier
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
SGD Classifier
NOT WORKING
kernel approximation
YES
number of categories known
YES
<10K samples
NO
KMeans
NOT WORKING
Spectral Clustering
GMM
NO
<10K samples
YES
Meanshift
VBGMM
NO
MiniBatch KMeans
clustering
YES
<100K samples
NO
SGD Regressor
YES
few features should be important
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
Lasso
ElasticNet
regression
YES
<10K samples
NO
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
kernel approximation
dimensionality reduction

Why KNN?

- Machine learning that uses a set of input values to predict output values
- We have the proper labeled data
- Small dataset
- One of the simplest models for classification

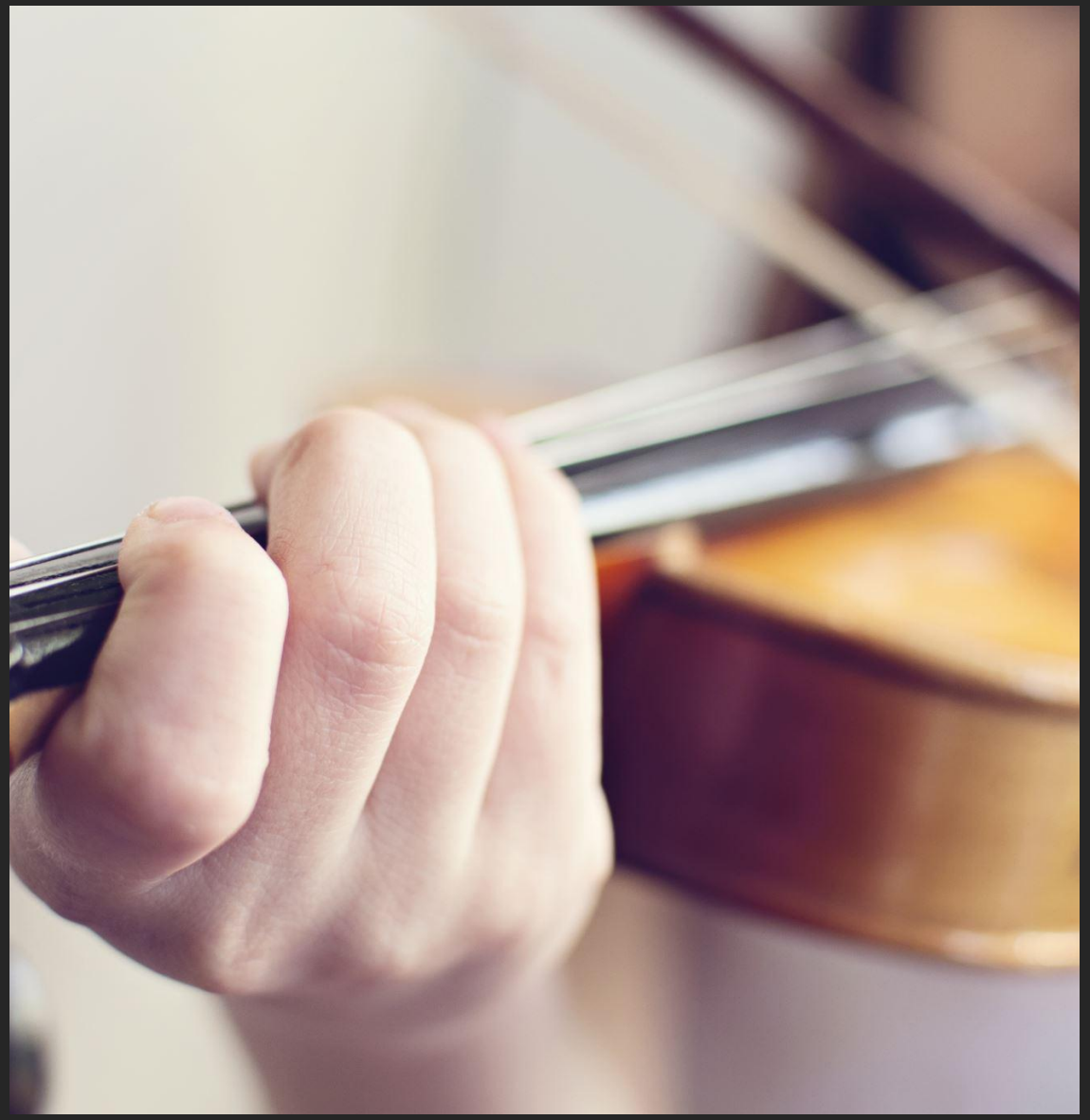


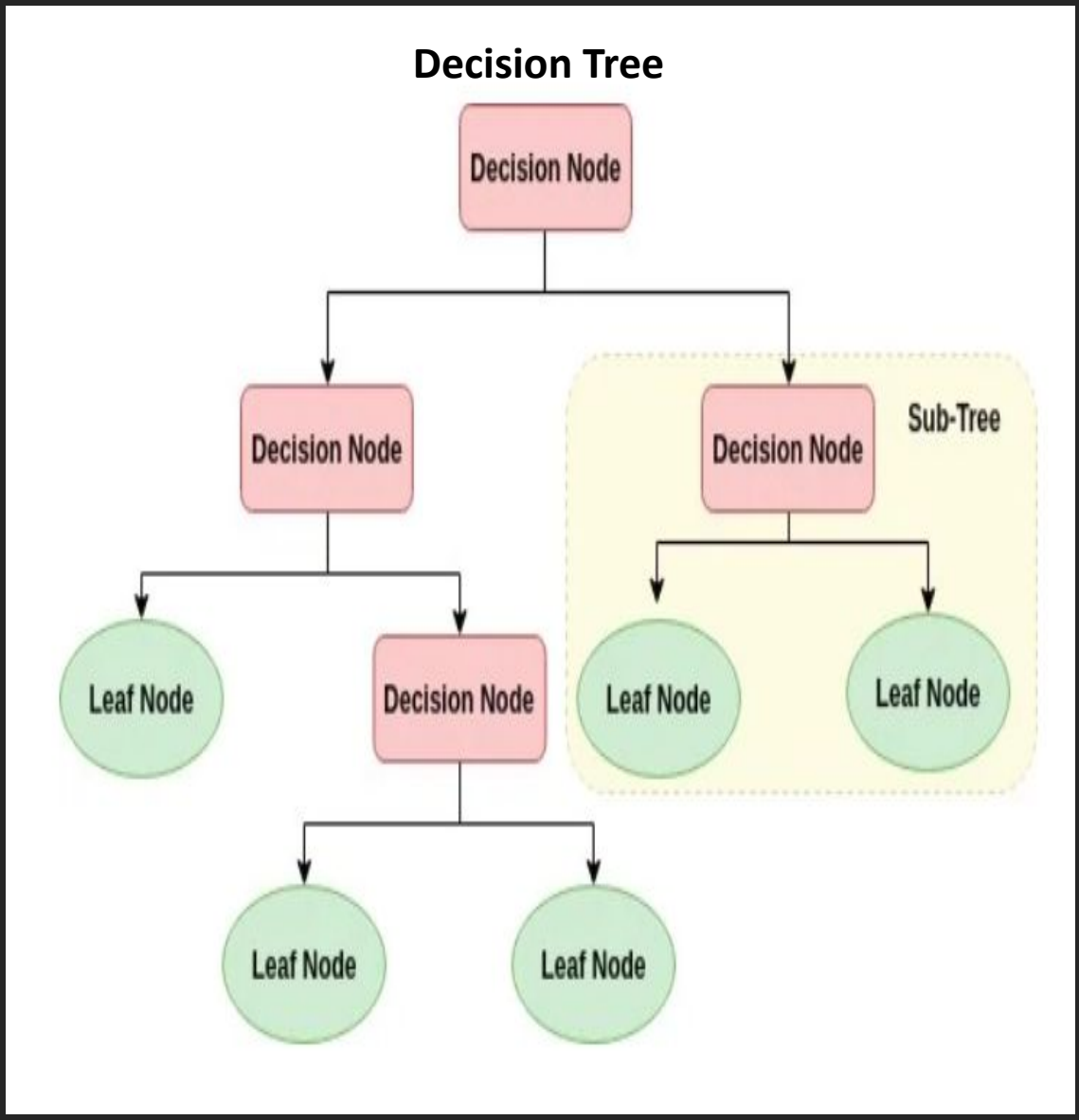
KNN Model



Why Decision Tree?

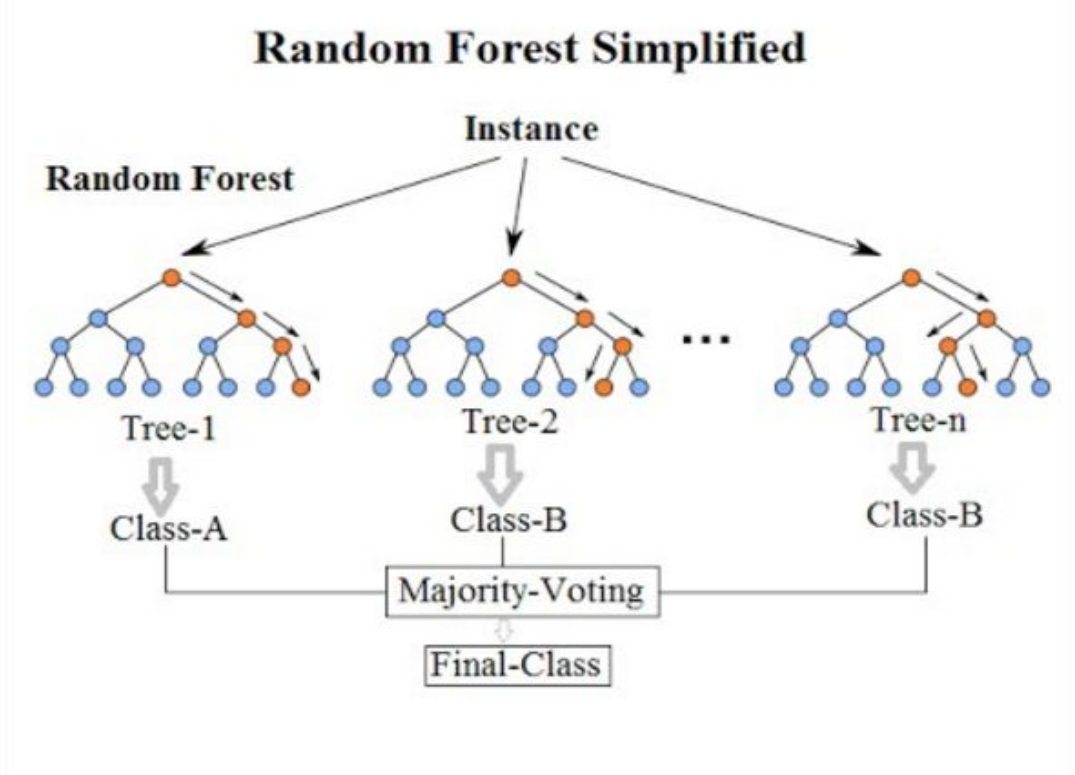
- Visual representation of a decision situation
- Show all those factors within the analysis that are considered relevant to the decision
- Captures the idea that if different decisions were to be taken then the structural nature of a situation
- Simple to understand




Decision Tree
Decision Node
Decision Node
Decision Node
Sub-Tree
Leaf Node
Decision Node
Leaf Node
Leaf Node
Leaf Node
Leaf Node

Why Random Forest?

- It can handle binary, categorical and numerical features
- Only needs a very little pre-processing
- Parallelizable and easy to deploy
- Great with high dimensionality
- Robust to outliers by binning them


Random Forest Simplified
Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

Outputs

```
TOP 100:
0    4375
1     100
Name: Top100, dtype: int64
TOP 500:
0    3992
1     483
Name: Top500, dtype: int64
TOP 750:
0    3520
1     955
Name: Top1000, dtype: int64
TOP 1000:
0    3520
1     955
Name: Top1000, dtype: int64
TOP 1500:
0    3047
1    1428
Name: Top1500, dtype: int64
TOP 2000:
```

Command took 0.05 seconds -- by rombale

```
X_train_100 = X_train_100[predictors].values
X_test_100 = X_test_100[predictors].values
X_train_500 = X_train_500[predictors].values
X_test_500 = X_test_500[predictors].values
X_train_750 = X_train_750[predictors].values
X_test_750 = X_test_750[predictors].values
X_train_1000 = X_train_1000[predictors].values
X_test_1000 = X_test_1000[predictors].values
X_train_1500 = X_train_1500[predictors].values
X_test_1500 = X_test_1500[predictors].values
X_train_2000 = X_train_2000[predictors].values
X_test_2000 = X_test_2000[predictors].values
```

Command took 0.04 seconds -- by rombaleynaud@student.eae.

Outputs

Grid Search methodology :

GridSearch method, it consists of trying a multitude of values for each hyperparameter and then selecting that which minimizes the error on the basis of the test.

```
##TOP1000##
#KNN MODEL#
#Creation of the dictionnary in preparation of GridSearch
param_knn = {'n_neighbors': [5, 10, 50, 100, 175, 200, 215, 225, 235, 250], 'weights': ['uniform', 'distance'], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']}
#fit Gridsearch on data
model_knn1000 = GridSearchCV(KNeighborsClassifier(), param_knn, refit=True, verbose=2)
model_knn1000.fit(X_train_1000,y_train_1000)
#DECISION TREE MODEL#
#Creation of the dictionnary in preparation of GridSearch
param_dct = {'max_leaf_nodes': [2, 5, 7, 9, 10, 15, 20], 'min_samples_split': [2, 3, 4, 5], 'criterion':['gini','entropy'], 'max_depth':[3,4,5,6,7,8,10]}
#fit Gridsearch on data
model_dct1000 = GridSearchCV(DecisionTreeClassifier(), param_dct, refit=True, verbose=2)
model_dct1000.fit(X_train_1000,y_train_1000)
#RANDOM FOREST MODEL#
#Creation of the dictionnary in preparation of GridSearch
param_rdf = {'n_estimators': [50], 'max_leaf_nodes': [2, 5, 7, 9, 10, 15, 20], 'min_samples_split': [2, 3, 4, 5], 'criterion':['gini','entropy'], 'max_depth':[3,4,5,6,7,8,10]}
#fit Gridsearch on data
model_rdf1000 = GridSearchCV(RandomForestClassifier(), param_rdf, refit=True, verbose=2)
model_rdf1000.fit(X_train_1000,y_train_1000)
```

Outputs

GridSearch results for the TOP750 configuration:

```
KNN Model
KNeighborsClassifier(n_neighbors=50, weights='distance')
```

```
Decision Tree
DecisionTreeClassifier(criterion='entropy', max_depth=4, max_leaf_nodes=15)
```

```
Random Forest Model
RandomForestClassifier(max_depth=10, max_leaf_nodes=20, min_samples_split=5,
                       n_estimators=50)
```

GridSearch results for the TOP1000 configuration:

```
KNN Model
KNeighborsClassifier(n_neighbors=50, weights='distance')
```

```
Decision Tree
DecisionTreeClassifier(criterion='entropy', max_depth=4, max_leaf_nodes=15)
```

```
Random Forest Model
RandomForestClassifier(max_depth=10, max_leaf_nodes=20, min_samples_split=5,
                       n_estimators=50)
```

Evaluation

Models		Top 100	Top 1000	Top 2000	Top 2500
KNN	Accuracy	0,98	0,77	0,64	0,62
	Precision	0,75	0,52	0,59	0,64
	Recall	0,13	0,178	0,54	0,78
	F1 Score	0,22	0,26	0,56	0,7
Decision Tree	Accuracy	0,98	0,76	0,56	0,6
	Precision	0	0	0,61	0,59
	Recall	0	0	0,15	0,98
	F1 Score	0	0	0,24	0,74
Random Forest	Accuracy	0,98	0,76	0,57	0,62
	Precision	0	0,33	0,58	0,61
	Recall	0	0,04	0,34	0,95
	F1 Score	0	0,08	0,43	0,74

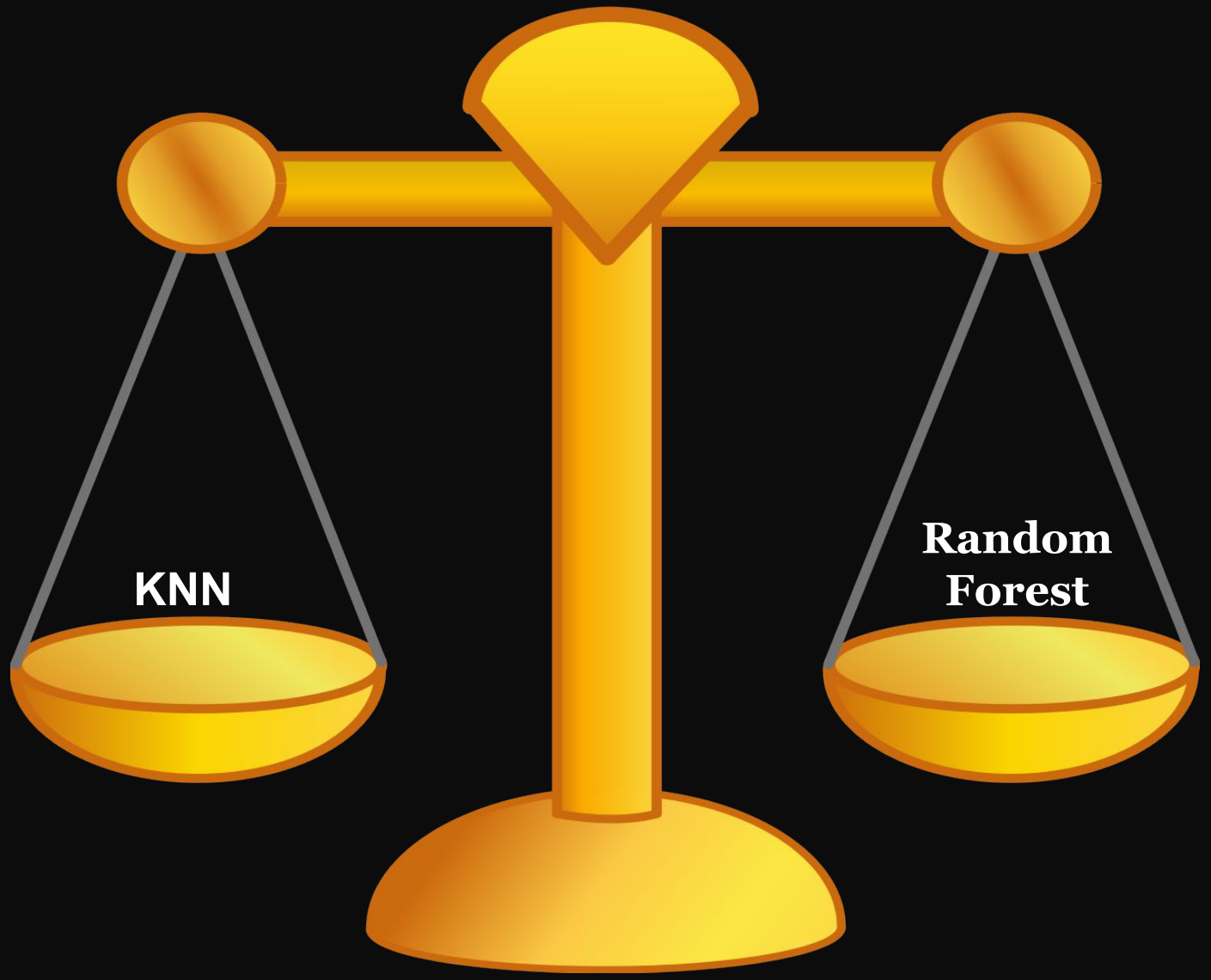
Accuracy = TP+TN/TP+FP+FN+TN

Precision = TP/TP+FP

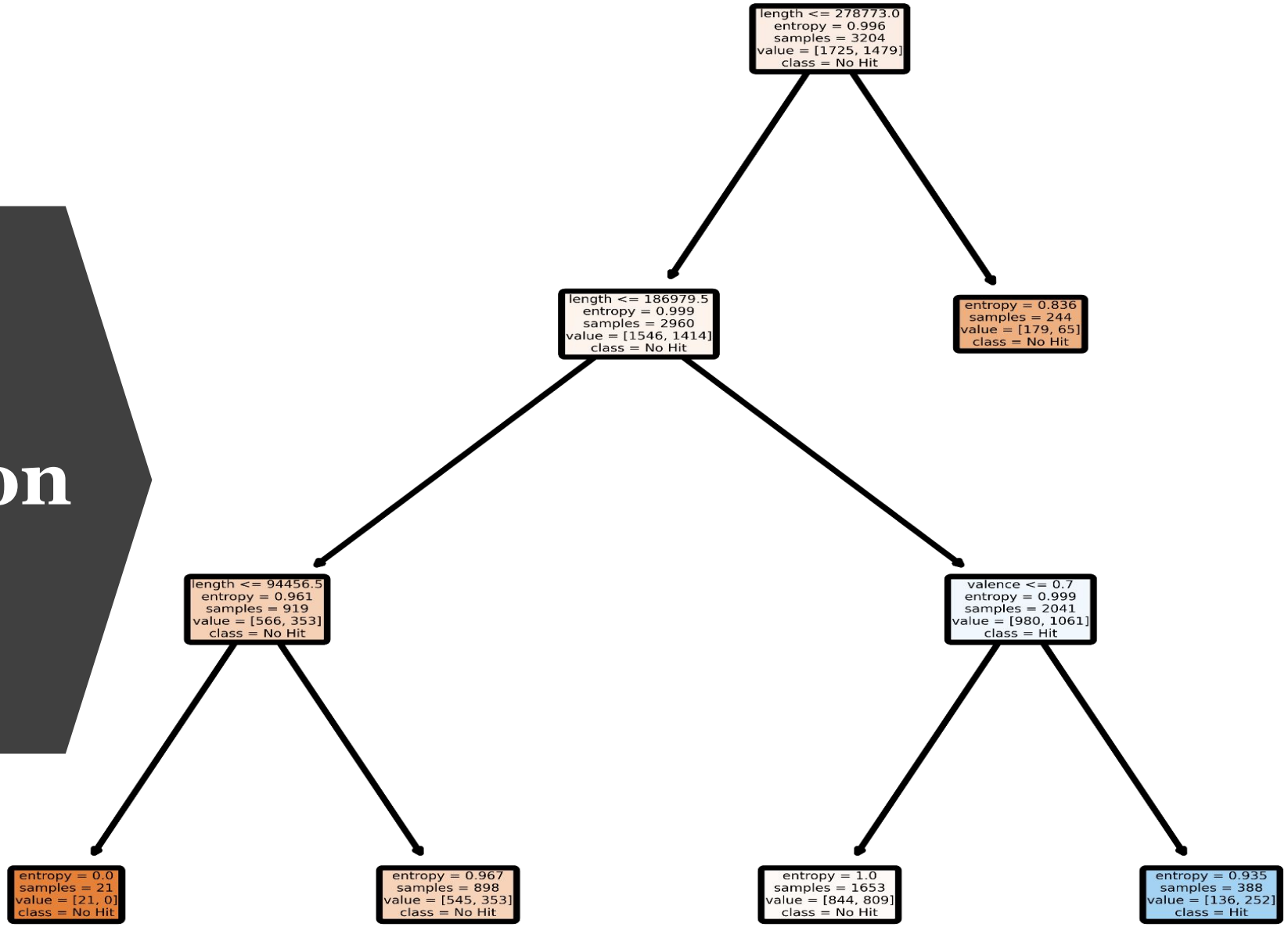
Recall = TP/TP+FN

False positives and False negatives aftermatch

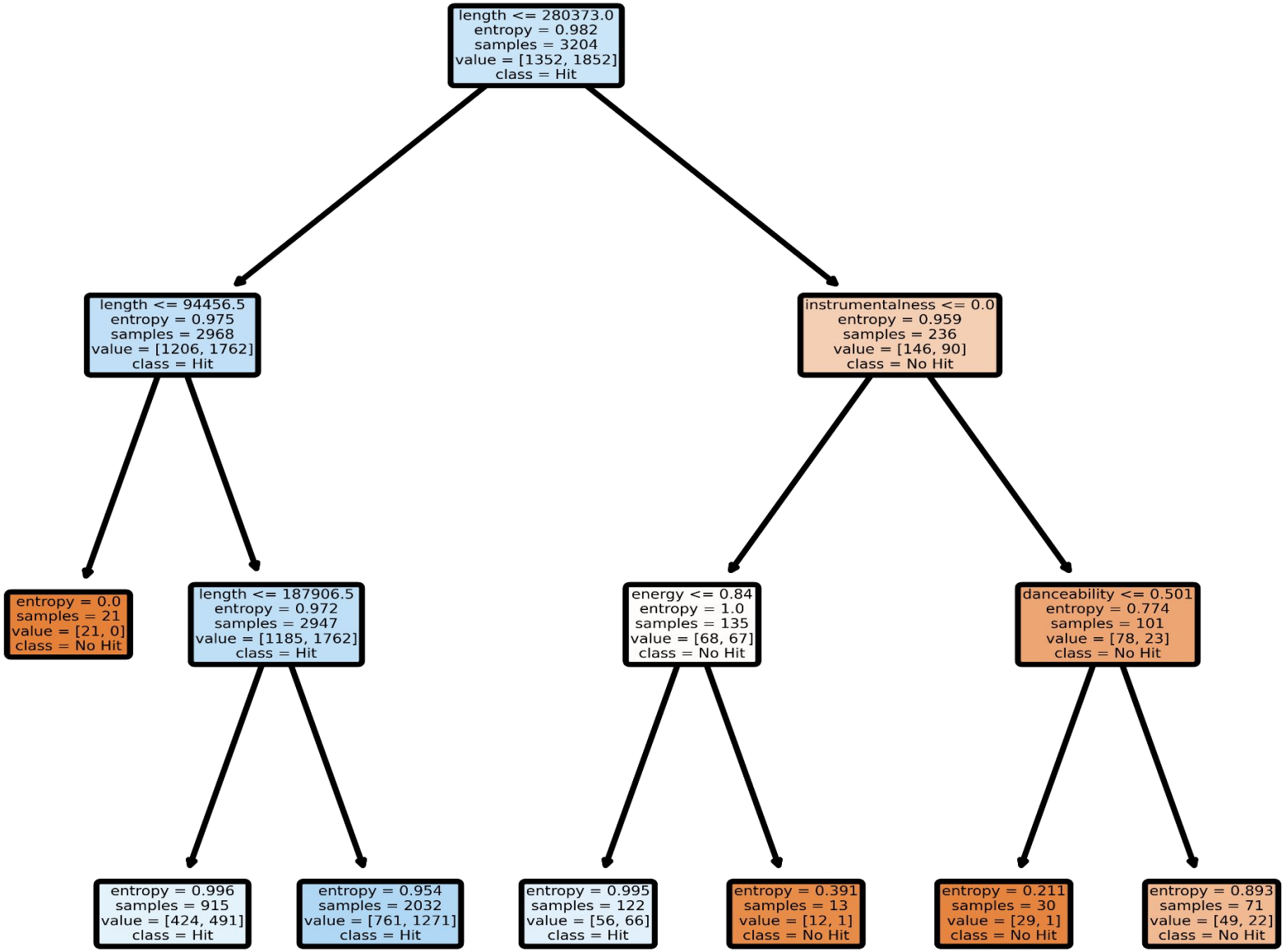
Evaluation



Evaluation


length <= 278773.0
entropy = 0.996
samples = 3204
value = [1725, 1479]
class = No Hit
length <= 186979.5
entropy = 0.999
samples = 2960
value = [1546, 1414]
class = No Hit
entropy = 0.836
samples = 244
value = [179, 65]
class = No Hit
length <= 94456.5
entropy = 0.961
samples = 919
value = [566, 353]
class = No Hit
valence <= 0.7
entropy = 0.999
samples = 2041
value = [980, 1061]
class = Hit
entropy = 0.0
samples = 21
value = [21, 0]
class = No Hit
entropy = 0.967
samples = 898
value = [545, 353]
class = No Hit
entropy = 1.0
samples = 1653
value = [844, 809]
class = No Hit
entropy = 0.935
samples = 388
value = [136, 252]
class = Hit

Evaluation


length <= 280373.0
entropy = 0.982
samples = 3204
value = [1352, 1852]
class = Hit
length <= 94456.5
entropy = 0.975
samples = 2968
value = [1206, 1762]
class = Hit
instrumentalness <= 0.0
entropy = 0.959
samples = 236
value = [146, 90]
class = No Hit
entropy = 0.0
samples = 21
value = [21, 0]
class = No Hit
length <= 187906.5
entropy = 0.972
samples = 2947
value = [1185, 1762]
class = Hit
energy <= 0.84
entropy = 1.0
samples = 135
value = [68, 67]
class = No Hit
danceability <= 0.501
entropy = 0.774
samples = 101
value = [78, 23]
class = No Hit
entropy = 0.996
samples = 915
value = [424, 491]
class = Hit
entropy = 0.954
samples = 2032
value = [761, 1271]
class = Hit
entropy = 0.995
samples = 122
value = [56, 66]
class = Hit
entropy = 0.391
samples = 13
value = [12, 1]
class = No Hit
entropy = 0.211
samples = 30
value = [29, 1]
class = No Hit
entropy = 0.893
samples = 71
value = [49, 22]
class = No Hit

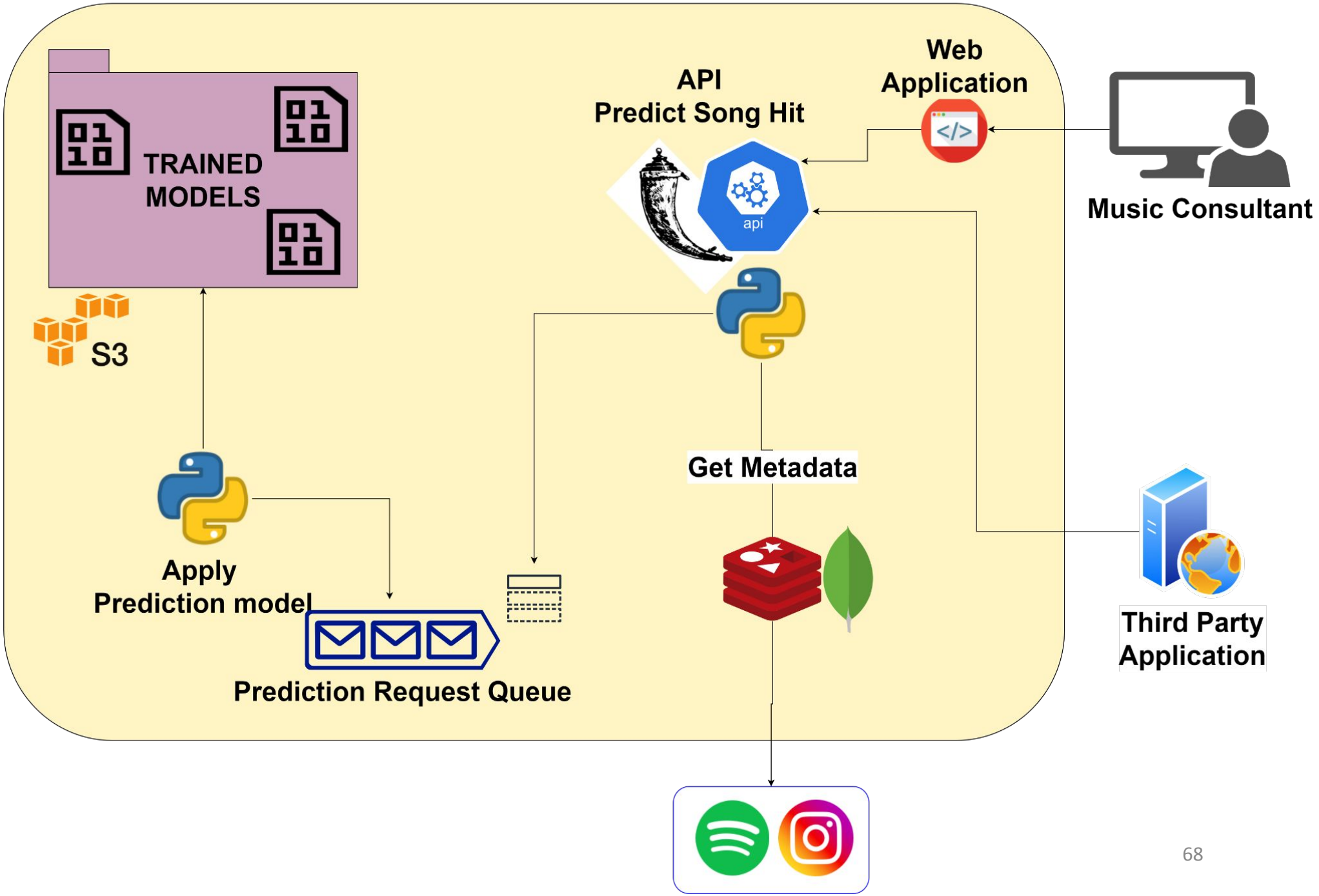
Deployment

- API-first
- Simple Responsive Web Application
- Manual monitoring and re-train

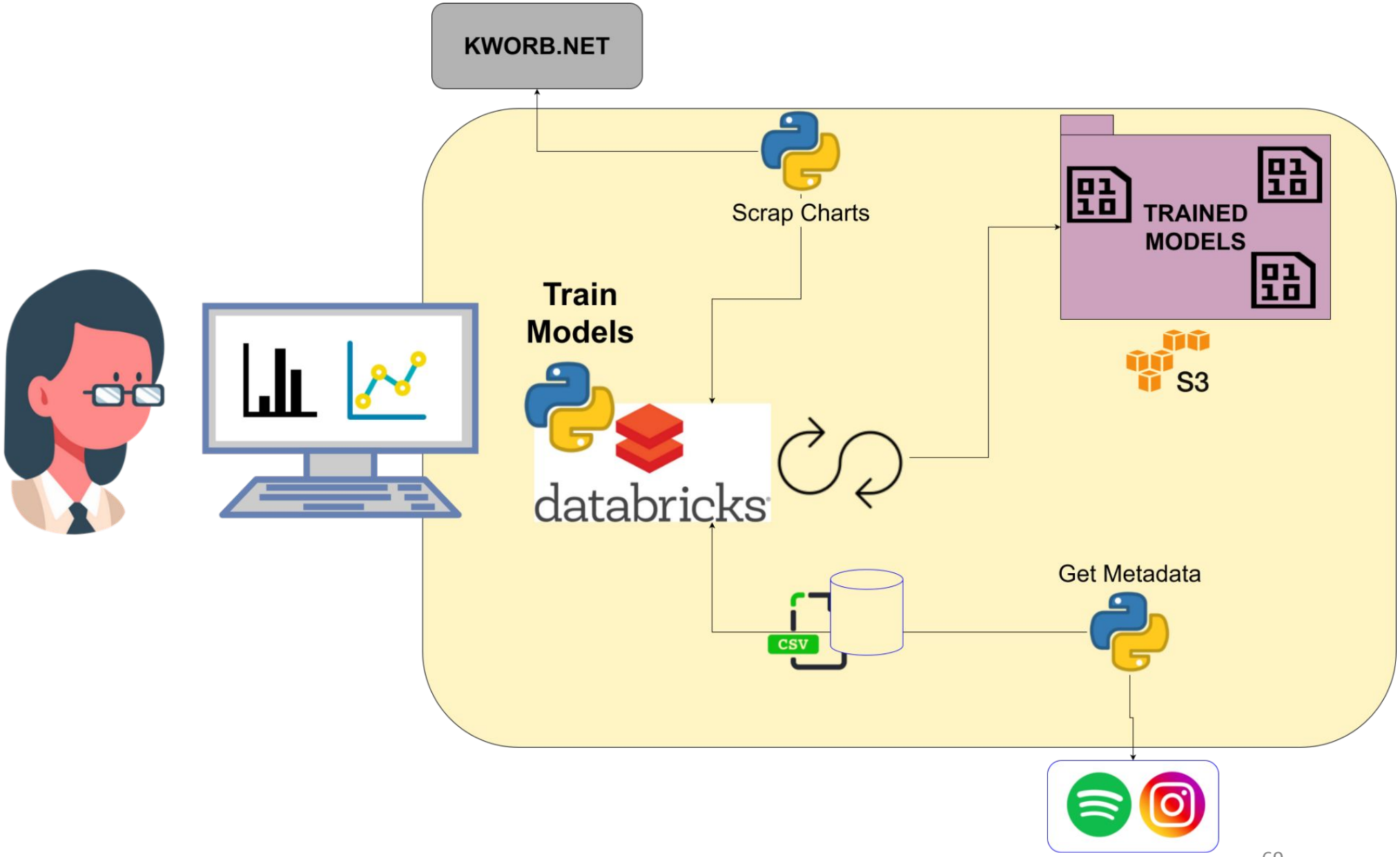
Tentative model update frequency:

- No less than 5 weeks
- At least every 15 weeks

Deployment


TRAINED MODELS
S3
API
Predict Song Hit
api
Web Application
Music Consultant
Get Metadata
Apply Prediction model
Prediction Request Queue
Third Party Application

Model Evolution



Conclusion

- Currently, the leg work has been done, the output is promising. But unfortunately, our model does not have good evaluation results.

Improvement for next iteration:

- Gather more songs to train the models

Models		Top 100	Top 1000	Top 2000	Top 2500
KNN	Accuracy	0,98	0,77	0,64	0,62
	Precision	0,75	0,52	0,59	0,64
	Recall	0,13	0,178	0,54	0,78
	F1 Score	0,22	0,26	0,56	0,7
Decision Tree	Accuracy	0,98	0,76	0,56	0,6
	Precision	0	0	0,61	0,59
	Recall	0	0	0,15	0,98
	F1 Score	0	0	0,24	0,74
Random Forest	Accuracy	0,98	0,76	0,57	0,62
	Precision	0	0,33	0,58	0,61
	Recall	0	0,04	0,34	0,95
	F1 Score	0	0,08	0,43	0,74

THANK YOU
EVERYONE
quickmeme.com