SoundsGood

Over the years, a multitude of songs have reached the top of the most famous music charts. We can see that each era has been marked by a musical trend.
Why does a song appeal to a specific audience ? Why do some songs never find success with the same audience ?
In this day and age, music is mainly consumed online and the music industry is investing more and more to understand online user data.
This large amount of data makes it possible to study the impact of an artist or a song and offer music titles to all types of users.
It is now possible to extract the probability that a song will appeal to a user using **SOUNDSGOOD.**
By reading this article, you will discover how **SOUNDSGOOD** was designed.

Step 1: the choice of the Dataset

We've selected the Spotify Song Attributes dataset from the Spotify's API. It contains data on the 2700 tracks that a user liked or disliked.
There are 16 different columns in this dataset, 13 of them are song attributes. There's a column for the song name, another column for the artist's name and a final column called *target* which is the label of the song : each one is labeled with a "1" when the user likes them and "0" for the ones he doesn't like.

Here are the 13 attributes of the song:

o *Tempo* indicates the rhythm of the song.
o *Energy* indicates songs that are perceived as fast, loud and noisy. For example, a rock song will score higher on the energy scale than a classical song.
o *Danceability* is the ability to dance to the song. When its score increases, it is easier to dance to that song.
o *Loudness* is the loudness of the song.
o *Liveness* indicates the presence of an audience during recording. The higher the liveness note, the more likely the song was played live.
o *Instrumentalness* is the measure that defines the use of instruments in music.

o *Valence* indicates the positivity felt when we listen to the song.
o *Duration_ms* is the duration of the song.
o *Acousticne*ss indicates whether the sound is acoustic.
o *Speechiness* detects the presence of spoken words in a song: the higher it is, the more spoken words are present.
o *Key* the estimated global music key of the song.
o *Mode* indicates the mode (major or minor) of a song.
o *Time_signature* is an estimate of the global rhythm signature (number of beats) of a track.

Step 2: Data Cleaning

Many features have similarities. For example, *instrumentalness* is the measure that defines the use of instruments in music, while *speechiness* measures the use of the voice as discussed above.
Both features therefore measure the musicality of a song.

We've created feature categories :

- <u>Class Labels</u> : *target*
- <u>Interval</u> : *acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence*
- <u>Time</u> : *duration_ms*
- <u>Numerical</u> : *tempo*
- <u>Ordinal</u> : *key, time_signature*
- <u>Binary</u> : *mode*
- <u>String</u> : *song_title & artist*

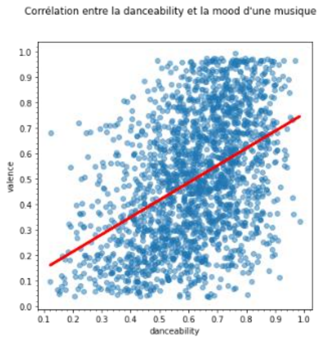
Using the <u>binning method</u>, we will create new features from the families we have created previously.

- instru_ind : The instrumental of a music and its lyrics
- beat_ind : The energy and the beat of a music
- melody_ind : All the melody of a music
- env_ind: The atmosphere of a music
- valence

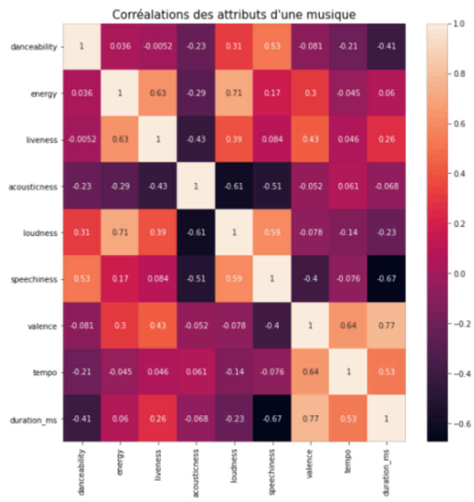
We also decided to clean our dataset to keep only the necessary data. To do this, we replaced the string characters in the song_title, artist columns with the value null.

Step 3: Datavisualization

This step allowed us to get an overview of our data and to understand the correlations between them. We performed a feature analysis in a graphical way.

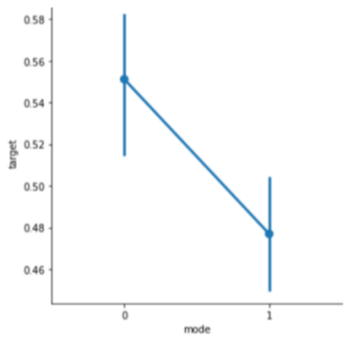


Thanks to the graph above, we can see that when the *danceability* increases the *valence* also increases : these two features are therefore correlated.



The Heatmap above shows the correlation between the attributes: there is a strong linear relationship between *loudness* and *valence* for example.

Finally, we analyzed the representative *target* curve according to the *mode.*



We can see that the more the *mode* (which indicates the modality (major or minor) of a song) increases, the *target* (which indicates whether the user likes the song or not) decreases.

Afterwards, it was important to study the value of the target according to the values of our features. We created value data intervals for each feature with the associated target value. We obtain the following results:

valence_band	target
(0.0338, 0.274]	0.423246
(0.274, 0.513]	0.490196
(0.513, 0.753]	0.539405
(0.753, 0.992]	0.579365

	tempo_band	target
0	(47.688, 90.727]	0.413043
1	(90.727, 133.595]	0.529218
2	(133.595, 176.463]	0.485323
3	(176.463, 219.331]	0.557377

loudness_band	
(-33.13, -24.9]	0.090909
(-24.9, -16.702]	0.220000
(-16.702, -8.504]	0.673660
(-8.504, -0.307]	0.470858

	liveness_band	target
0	(0.0178, 0.256]	0.499348
1	(0.256, 0.494]	0.517808
2	(0.494, 0.731]	0.510870
3	(0.731, 0.969]	0.692308

	acoustic_band	target
0	(-0.000992, 0.249]	0.531187
1	(0.249, 0.498]	0.496154
2	(0.498, 0.746]	0.546218
3	(0.746, 0.995]	0.231293

	energy_band	target
0	(0.0138, 0.261]	0.152381
1	(0.261, 0.506]	0.583333
2	(0.506, 0.752]	0.555270
3	(0.752, 0.998]	0.480460

	dance_band	target
0	(0.121, 0.338]	0.469565
1	(0.338, 0.553]	0.377820
2	(0.553, 0.769]	0.506442
3	(0.769, 0.984]	0.703601

	speech_band	target
0	(0.0223, 0.221]	0.481298
1	(0.221, 0.42]	0.740113
2	(0.42, 0.618]	0.650000
3	(0.618, 0.816]	0.500000

	instru_band	target
0	(-0.000976, 0.244]	0.467635
1	(0.244, 0.488]	0.724138
2	(0.488, 0.732]	0.702128
3	(0.732, 0.976]	0.644809

We thus have a vision of the indicators that allow the target score to increase and those that make it decrease.

Step 4: our prediction algorithms

We will base our work on the following three prediction models:

- Logistic regression which is used to model the probability of the existence of a certain class or event. In our study, this algorithm will allow us to model the probability that a song will appeal to a user.
- Decision Tree Classifier uses a decision tree to move from observations about an item to conclusions about the target value of the item.
- K-NN is an algorithm that consists of choosing the k data closest to the studied point in order to predict its value.

The results are as follows:

Training test

```
1. Logistic Regression:  65.98
2. Decision Tree Classifier:  89.23
3. K-NN:  74.27
```

Result test

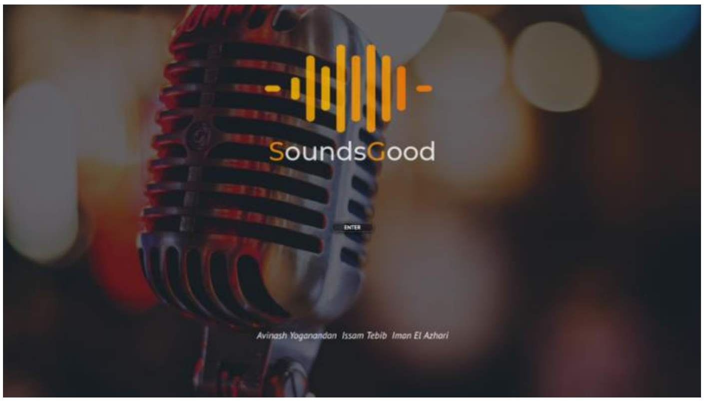
```
1. Logistic Regression: 61.39
2. Decision Tree: 61.06
3. K-NN: 61.22
```

K-Kross validation

```
1. Logistic Regression:  57.38
2. Decision Tree Classifier:  57.07
3. K-NN:  60.22
```

Step 5: Creating the SoundsGood interface

We have created a fun and easy to use interface to which we have implemented our prediction model. Thus, we made a React application that we deployed on Heroku.



Thanks to the SoundsGoods application, you can find out in a few clicks the potential of a song to be appreciated by a user.
First of all you have to enter the title of the song of our choice in the dedicated field and then the plausibility that the song will be liked by our user will be returned according to the different prediction models we have worked with.

Join us soon on SoundsGood!