

Product Vision

The why, who and what

MoodCat.me

April 28, 2015

1 Why should we build MoodCat?

Various competitors provide a system that can offer users popular songs or songs that are similar to songs the user already listened to. These recommendations mostly consist of songs that are popular or users with similar preferences listened to. This leaves a huge amount of music produced by a lot of artists undiscovered.

Secondly, most music platforms do not offer user interaction. Some competitors offer users rooms to socialize while experiencing songs. An example is Plug DJ [5] where users can create a room and listeners can suggest Youtube-songs [7] for the playlist. Such a playlist has to be maintained in order to keep running and therefore does not suit the listeners that want to listen on their own. Furthermore, it has to be moderated to prevent unwanted suggestions that might not be interesting for the listeners.

Users want to discover music and have social interaction. MoodCat allows users (and friends!) to unite in rooms, and listen to and discuss music together. Hereby we apply a successful feature from the live video/gaming streaming industry (for example TwitchTV [6]) to the music industry. Music rooms never fall silent, instead they are continuously filled with relevant music that fits the mood of the listeners in the room.

2 What is the target audience of MoodCat?

Our target audience consists of (but not limiting to) the following persons:

1. Everyone who wants to listen to music without the hassle of searching suitable songs for both their musical preference and current mood.
2. Everyone who wants to discover and listen to songs they have not heard yet.
3. Everyone who wants to listen to and experience songs and share this experience with fellow listeners in the same online room.
4. Everyone who wants to listen at any time on the day, no matter how many users are currently online.

3 What are the selling points of MoodCat?

MoodCat has several selling points that would make it unique in the current music market.

1. MoodCat is self-maintaining, which makes it possible to handle audience sizes from one to one thousand. This makes it possible to use the service no matter how many other users are currently online.
2. MoodCat offers users both to experience solely or communicate with fellow music listeners.
3. The system already works without user input as specific interpretation of audio features allow us to find similar songs.
4. The system is able to offer relevant songs using the specified current mood of the user.
5. With MoodCat unknown or lesser known artists have equal chances to let their songs be played. This is a result of the system choosing songs based on features and is therefore unbiased regarding play counts.
6. MoodCat can determine mood trends of a long-term user and can adjust the room-ordering mechanism accordingly. Therefore long-term users are served more suitable rooms which increases their overall satisfaction.
7. MoodCat can suggest users who are repeatedly not satisfied with the songs played in a room, a different and more suitable room.

4 Which components should MoodCat have?

MoodCat will be developed as three separate components: the frontend, the backend and a special music matching algorithm.

1. The frontend will be the visible part for the listener. After logging in to our services, the user will be able to select his mood and based on that MoodCat will provide several music listening rooms. In these rooms the user can interact with other listeners through the chat while listening to the music.
2. The backend keeps track of which rooms are active and what song is currently played in the room. Furthermore, the backend is responsible for transmitting the chat messages between the users in the room.
3. The music matching algorithm is used to compare and match moods of songs and chatrooms. It has two roles: list the room suggestions for a user and provide the next song to be played for a room.

4.1 Frontend

The MoodCat frontend will be web-based. This allows us to support all modern devices which enables a large target audience. We will develop the frontend using modern techniques as HTML5 [3], CSS3 [2] and AngularJS [1]. Users can login using their Soundcloud account. This allows us to integrate with the Soundcloud platform and prevents having to register for our services first.

4.2 Backend

The MoodCat backend will be a RESTful [10] API written in Java using JAX-RS. It will be backed by a relational (PostgreSQL) database. The backend connects the frontend to the music matching algorithm.

4.3 Music matching algorithm

The Music matching algorithm determines and compares moods for songs, rooms and users. Moodcat will be developed around a learning algorithm. This means that Moodcat will improve over time. Default values will be determined by examining the audio features provided by our data set. Then, through a system of user votes, the system will learn and improve over time.

We discovered that moods and emotions can be mapped to a valence/arousal space (see figure 1).

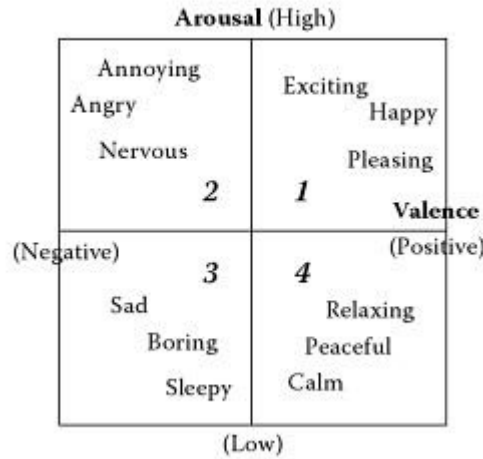


Figure 1: The valence/arousal graph [8]

Arousal and valence are physiological terms. Arousal means how calm or excited you are. Valence is the amount of attraction you have towards a certain event.

In order to use arousal and valence, we have to express them in terms of music features. For valence we can use harmony, mode and tonality. For arousal we can use pitch, timbre, speed and loudness [4] [9].

The data of EEMCS does not explicitly provide the above features. However, we can extract some of them from the low-level features the data does provide. The low-level features we can use for valence:

1. For harmony we can use the consonance and dissonance of a song.
2. To determine the mode of a song, we can use the key, key scale and key strength from the data. The key strength also tells a bit about the tonality of the song. When the key strength approaches 1, we can speak of tonality. When the key strength approaches 0, we can speak of atonality.

The low-level features we can use for arousal:

1. Beats-per-minute (BMP) gives us the tempo of a song.
2. Loudness can be directly retrieved from the loudness feature.
3. Pitch can be derived from the tuning frequency, which is standardized at 440 Hz but may vary depending on music genre, era and geography.

The timbre of a song is a separate research field and would be too difficult to integrate into our system.

References

- [1] AngularJS. <http://www.angularjs.org>. Accessed: 2015-04-28.
- [2] CSS. <http://www.w3.org/Style/CSS/>. Accessed: 2015-04-28.
- [3] HTML. <http://www.w3.org/TR/html5/>. Accessed: 2015-04-28.
- [4] Music Emotion Recognition presentation. <http://www2.ensc.sfu.ca/~ljilja/cnl/guests/Music.pdf>. Accessed: 2015-04-28.
- [5] plug.dj join the party. <https://plug.dj>. Accessed: 2015-04-28.
- [6] Twitch. <http://twitch.tv>. Accessed: 2015-04-28.
- [7] YouTube. <https://youtube.com>. Accessed: 2015-04-28.
- [8] Y. Chang and H.H. Chen. *Music Emotion Recognition*. Boca raton, State: CRC Press, 2011.
- [9] E. Coutinho and A. Cangelosi. Musical emotions: Predicting second-by-second subjective feelings of emotion from low-level psychoacoustic features and physiological measurements, 2011.
- [10] Roy Thomas Fielding. *REST: Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- [11] Karin Kosina. Music genre recognition. Master's thesis, Fakultät für Informatik, Kommunikation, Medien Hagenberg, 2002.
- [12] Matt Mcvicar, Yizhao Ni, Tijl De Bie, and Raul Santos-rodriguez. Leveraging noisy online databases for use in chord recognition, 2011.