Storytape

A Project by Franch & Ingberg

Dataset link:

The dataset was of our own creation, via scrapping with Spotify's API. For the purpose of the projects we made it public and uploaded it to http://www.kaggle.com/alexing/spotify-tracks.
The text datasets are available on the links provided at the text dataset section.

Problem to solve:

The creation of a certain playlist which matches the mood of a certain text. This text could be, but definitely not restricted to, a novel, a story, a movie script, a historical article, a biography, etc.
The ideal client would be anyone interested in using the app; mostly people who like to accompany their leisure reading with background music.

The client you'd like to solve the problem for:

People who like listening to music to set the mood for their readings. And also to companies that might want to sell e-books and music in a combined product.

High level solution description:

The user will provide an e-book in a text format as input to the webapp. The tile will be searched in our book database to get a genre tag. From the book genre, a set of musical genres is returned to set the general mood of the playlist. The literary and musical genres will be manually related and stored in a table.
The next step is to break the text in blocks with size related to the average reading speed. Each of these blocks will be used as input in our model which will output emotion tags for that chunk of text. These emotion tags will also be manually related to the Spotify features. We will provide these tags as inputs and get a set of Spotify features and then look in the database for a song with similar features and also being of the musical genres obtained in the previous step. This song will then be added to a playlist.
This process is repeated for each of the text blocks until a playlist for the whole text is created. The webapp will then serve an e-book reader with a Spotify player on the side and the song for each block will be played once the reader reaches them, setting a mood and soundtrack for the book.

Milestone plan:

- Creating an interesting track database via scrapping Spotify.
- Creating or finding an interesting text mood analyzing database.
- Decide on a Machine Learning strategy or model to analyze the text
- Decide which would be the criteria for mood analyzing. (Angry/Disgusted/Happy/Sad/Scared/Surprise? Happy/Sad? More complex stuff like “Epic”, “Nerdy”, “Floral”, “Fragile”?
- Creating a working model for analyzing a chunk of text and getting mood outputs.
- Creating an interface to match this outputs with our track database.
- Deciding on a criteria to decide the length of playlists and how to divide the chunks of the text so as to match them with tracks.
- Developing a process to create the playlists in Spotify and add them to some user’s profile.
- Integrating everything into a web application.
- Serving the app
- Being happy and proud with our product

Description of deliverables:

- Databases with data insights and exploration
- Description of models and techniques that’ll be used
- POC (pt. 1) with each part of the data workflow explained: a text matched with a playlist.
- POC (pt. 2) a working app (without UI or anything fancy) which receives a text as input and creates a spotify playlist.
- A working web application

Spotify Database:

Using Spotify’s WEB API[1] with Spotify[2] as a wrapper, we managed to pull a considerable dataset from their main db. We needed to do this since Spotify’s API doesn’t allow you to search for tracks using values for mood; the way it works is you ask for a track and then you have the option of knowing its values.
Approximately 450k registries of songs are useful if we discard duplicates. To make it the sample the more representative and unbiased we could, we divide the scrapping in 10 genres we considered the most popular: jazz, metal, pop, folk, rap, rock, R & B, country, latin, and

[1] https://developer.spotify.com/web-api/
[2] https://github.com/plamere/spotipy

electronic. We searched for each term and saved the first 50k results adding in total to half a million registers.

The useful features we find in each song are as follows[3]:

Audio Features Object		
Key	**Value Type**	**Value Description**
acousticness	float	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
danceability	float	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.
duration_ms	int	The duration of the track in milliseconds.
energy	float	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.
id	string	The Spotify ID for the track.
instrumentalness	float	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 instrumentalness 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.
key	int	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.
liveness	float	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.
loudness	float	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.
mode	int	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.
speechiness	float	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 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe

[3] https://developer.spotify.com/web-api/get-audio-features/

		tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
tempo	float	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.
time_signature	int	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).
uri	string	The Spotify URI for the track.
valence	float	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).

Our dataset also comprises the fields ‘name’, ‘artist’ and ‘genre’.

With this coefficients and labels we can pin down pretty accurately the general mood or feeling that a song conveys.
A data analysis with insights is attached in the form of a Jupyter Notebook[4].

Eventually, the Spotify API can be used to create the playlists directly into user’s profile while using each track’s Spotify ID.

Book and Text Datasets:

There are some public datasets available on GitHub[5][6] containing information about books such as title, author, genre, ISBN, etc. These data can be used to get the general information about the book the user provides as input and set a general tone for the soundtrack. For example, a book with the fantasy tag could have a classical and orchestral playlist while one with the horror tag might be better suited with some heavy metal. For this goal we can manually create the relationships between book and musical genres as those are not a large amount of tags.

For the main challenge of categorizing the emotion in blocks of text, we use the dataset EmoBank[7], which has 10k sentences tagged with values of valence, arousal and dominance. This data is going to be used to train a model (still to be determined) which will predict the general mood of a block of text. These emotions will then have to be related to the features present in the Spotify API so given a text an equivalent set of Spotify features can be obtained and songs with values close to these can be retrieved from the database.

[4] http://jupyter.org/
[5] https://github.com/akshaybhatia10/Book-Genre-Classification
[6] https://github.com/alexsanjoseph/goodreads-list-properties
[7] https://github.com/JULIELab/EmoBank