Million Song Project

Group 3, Milestone 2: EDA

Our TF is Evan MacKay and we've been in touch via email.

Project statement

The project aim is to predict the song hotness - or 'song hotttnesss' [sic] - based on its features (tempo, loudness, segment data etc), and metadata (artist information, year of release etc).

Project background and literature review

This project is based on the Million Song Dataset (MSD) provided by labROSA at Columbia University.

The majority of record company profits are made from smash hits, and a great deal of money and effort is spent on seeking out the next success. Thus, predicting song hotness would be a useful way to test new songs and predict their commercial potential. To our knowledge, this aspect has not been extensively explored in the literature– previous analysis using the MSD has concentrated on predicting year, genre and providing recommendations to users.

Previous attempts to predict hit songs have had mixed success, possibly due to the unpredictability of cultural markets[1]. Although there are industry rules of thumb on how to write hit songs, there is comparatively little in the machine learning space to automatically detect hits. Early attempts used acoustic and lyric-based features to build support vector machines have not resulted in reproduceable successful models, which is explained by the fact that these features are not informative enough[2].

More advanced features that have been extracted in the MSD may provide a better basis for song hotness prediction and have been used in a small number of studies to predict song popularity. One of the most relevant is Pham et al[3], which used song characteristics consisting of both acoustic features and metadata from MSD. The authors applied a number of machine learning algorithms (SVMs, neural networks, logistic regression, Gaussian discriminant analysis, and linear regression) to predict whether a song is popular or not. When designed as a classification task, the best models achieved around 80% test set accuracy compared to a 75% baseline (labelling only the top 25% of songs as being

[1] Salganik et al (2006) Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market https://www.princeton.edu/~mjs3/salganik_dodds_watts06_full.pdf

[2] Herremans et al (2014) Dance Hit Song Prediction http://antor.uantwerpen.be/wordpress/wp-content/papercite-data/pdf/herremans2014dance.pdf

[3] Pham et al (2015) Predicting Song Popularity http://cs229.stanford.edu/proj2015/140_report.pdf

popular). The best regression model achieved average error of 0.134 compared to a baseline of 0.159.

Companies like Spotify are using techniques[4] such as Collaborative Filtering[5], which can be relevant for our project as well, if combined with the right models. Collaborative Filtering attempts to make predictions about the interests of a user by collecting preferences or taste information from many users. The underlying assumption is that if a person agrees with another on one song, they will most likely also agree on other songs. The below mentioned section regarding «Taste Profile information» may be key in this regards.

Another resource that may be valuable for our research is the Kaggle competition[6] regarding this dataset. The winning entry based their models[7] on a paper by F. Aiolli[8], which may be another source of inspiration for us. The winning entry used an item-based Collaborative Filtering approach modified by 3 factors; 1) a parametric similarity between songs, 2) calibration of the scores, and 3) ranking aggregation with a user-similarity based predictor.

Preliminary EDA

The full dataset of a million songs is around 270 GB. There is also a subset of 10,000 songs (around 2.5 GB) that we will use for the purposes of developing the prediction approach, as indicated in the project guidelines. Both datasets are provided as compressed HDF5 files.

In the main dataset are audio analysis features for each track, including the target variable of song hotness. This has values ranging from 0-1, and indicates the popularity of a song. It is assigned algorithmically based on various metrics, including news/blog mentions, play counts, music reviews, radio airtime and Billboard rankings.

There are also an additional 53 variables that include:

- Several **artist and track IDs** that can be used for linking with other datasets: The Echo Nest[9] ID, Music Brainz, 7digital and playme.com.
- **Echo Nest tags**, which include 'artist terms', 'similar artists'.
- **Music Brainz tags** include the fields 'year', 'artist mbtags' and 'artist mbtags count'. These tags are applied by humans and there are fewer of them compared to the Echo Nest ones, but where they do exist they tend to be cleaner.
- **Artist data,** such as location, 'hotttnesss' and familiarity
- **Song data,** such as tempo, key, duration, title
- **Track analysis data,** containing arrays of features that correspond to different segments or sections of the track. The main acoustic features are pitches, timbre and loudness, as defined by the Echo Nest Analyze API. The API provides these for

[4] Answer provided by Erik Bernhardsson, responsible for machine-learning at Spotify, 2008-15: https://www.forbes.com/sites/quora/2017/02/20/how-did-spotify-get-so-good-at-machine-learning/#1f522c2b665c

[5] Hu, Koren and Volinsky - Collaborative Filtering for Implicit Feedback Datasets, http://yifanhu.net/PUB/cf.pdf

[6] https://www.kaggle.com/c/msdchallenge

[7] https://github.com/tuzzeg/solutions/tree/master/kaggle/msdchallenge

[8] Fabio Aiolli et al A Preliminary Study on a Recommentder System for the Million Songs Dataset Challenge - http://www.ke.tu-darmstadt.de/events/PL-12/papers/08-aiolli.pdf

[9] Now owned by Spotify

every “segment”, which are generally delimited by note onsets, or other discontinuities in the signal. The API also estimates the tatums, beats, bars (usually groups of 3 or 4 beats) and sections.

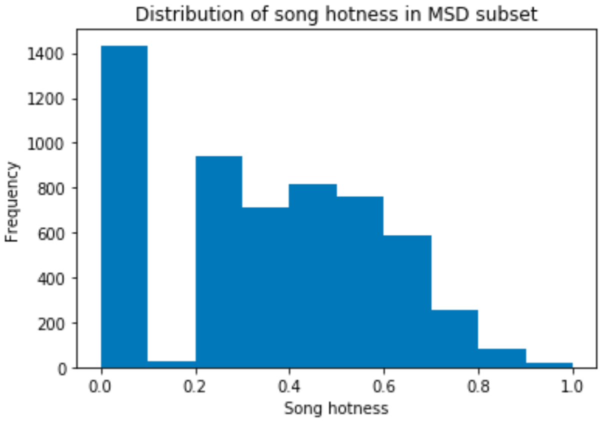
In terms of the data quality, we identified the following:

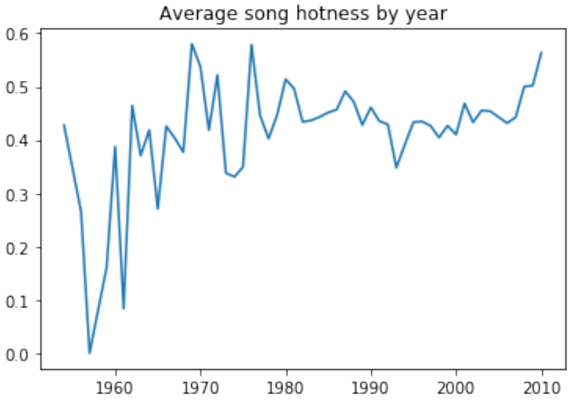
- `Song_hotttnesss` is missing for 43% of the rows in the subset. If these are removed, around 25% of the remaining rows have hotness values of 0
- Several columns are non-informative and can be dropped:
  - `Analysis sample rate:` is the same value for all tracks
  - `Energy` and `Danceability`: contain only zeros
  - `Artist latitude` and `longitude`: have more than 60% missing data
- `Year` has around 25% of rows with zero, which can be considered missing data. These are left as they are for now.

The data is accompanied by additional files that provide the following information:

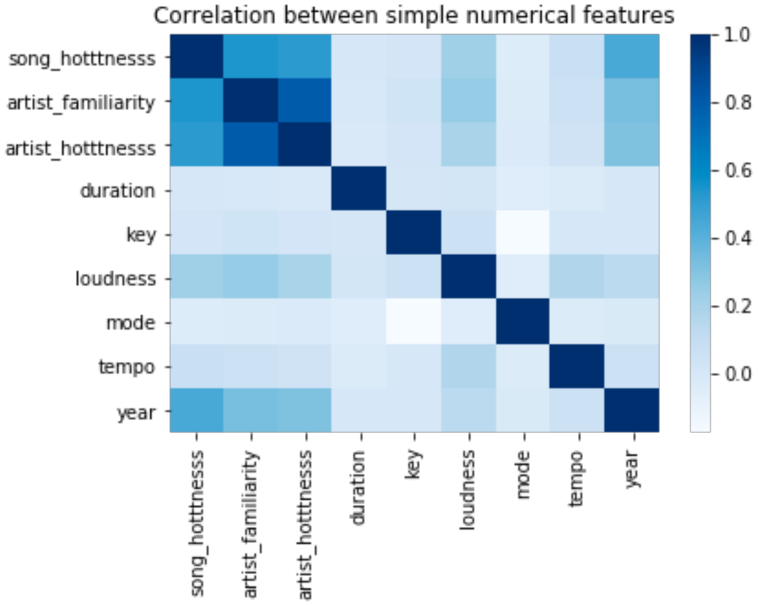
- List of track Echo Nest IDs
- List of all artist IDs
- List of all unique artist terms
- List of all unique musicbrainz tags
- List of the tracks for which there is year information
- List of artist latitude and longitude, where available
- Summary file of the whole dataset containing metadata but no audio analysis
- SQLite database containing track metadata
- SQLite database containing links from artist ID to tags
- SQLite database containing similarity among artists

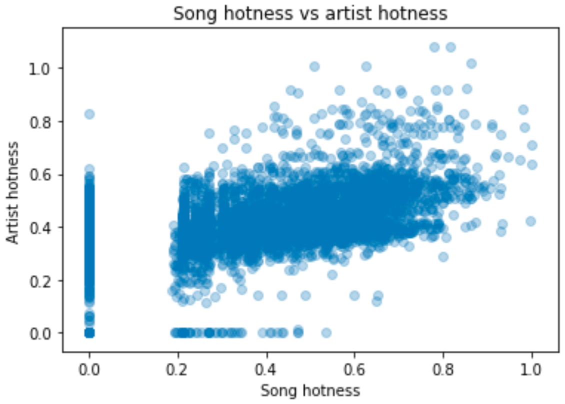
The distribution of song hotness is skewed with around 25% of the tracks having zero values (of the 5648 that do not have NaN values). Average song hotness per year seems to have a slight upward trend. We must also consider that musical tastes change over time.

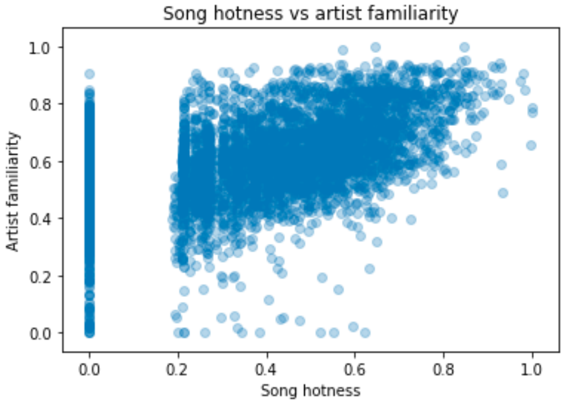




Exploration of correlations with the single-valued numerical columns shows that artist familiarity and artist hotness are strongly correlated with the target variable. As previously identified, there seems to be a positive correlation between song hotness and year, as well as with loudness.







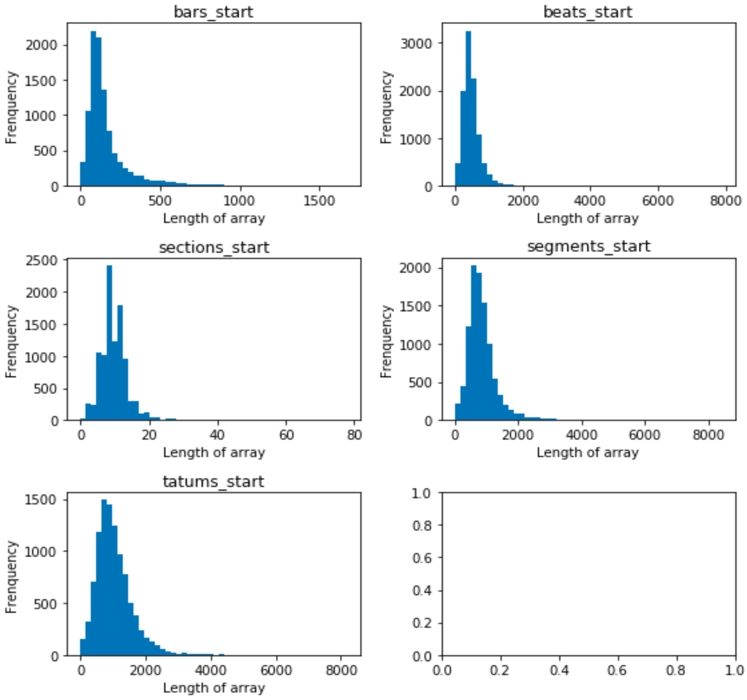
Several of the columns also contain numerical arrays of data for distinct parts of each song:

1. **Bars**: A bar (or measure) is a segment of time defined as a given number of beats.
2. **Beats:** A beat is the basic time unit of a piece of music; for example, each tick of a metronome. Beats are typically multiples of tatums.
3. **Sections:** Sections are defined by large variations in rhythm or timbre, e.g. chorus, verse, bridge, guitar solo, etc.
4. **Segments:** A set of sound entities (typically under a second) each relatively uniform in timbre and harmony.
5. **Tatums:** Tatums represent the lowest regular pulse train that a listener intuitively infers from the timing of perceived musical events (segments).

All of these parts have two columns associated with them:

- Start: the markers explained above, in seconds.
- Confidence: indicates the reliability of its corresponding attribute. Elements carrying a small confidence value should be considered speculative.

It is already clear from the descriptions that the arrays will be of different lengths since the length of the tracks and frequency of the measures will vary. Below are histograms showing the frequency of array lengths for the songs in the subset.

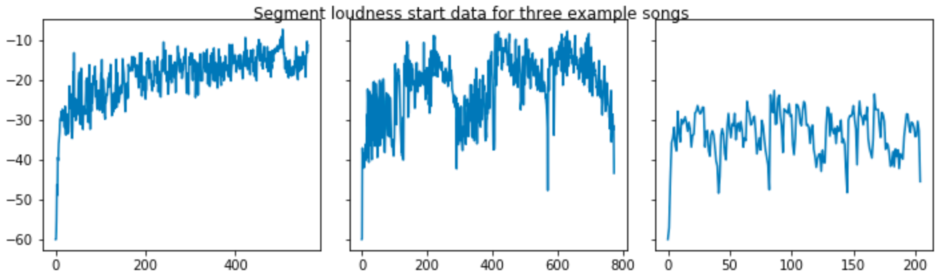


The most detailed information is for segments, which contain several other features:

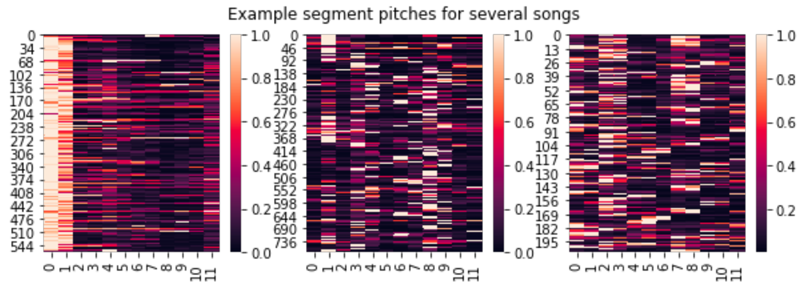
- Loudness_max: peak loudness value within the segment
- Loudness_max_time: offset within the segment of the point of maximum loudness
- Loudness_start: loudness level at the start of the segment
- Pitches: is given by a “chroma” vector, corresponding to the 12 pitch classes C, C#, D to B, with values ranging from 0 to 1 that describe the relative dominance of every pitch in the chromatic scale.
- Timbre: is the quality ('colour') of a musical note. It is what makes a particular musical sound different from another, even when they have the same pitch and loudness. The dataset represents timbre as a 12 dimensional vector of 'MFCC-like' features, each of which are unbounded values roughly centered around 0.

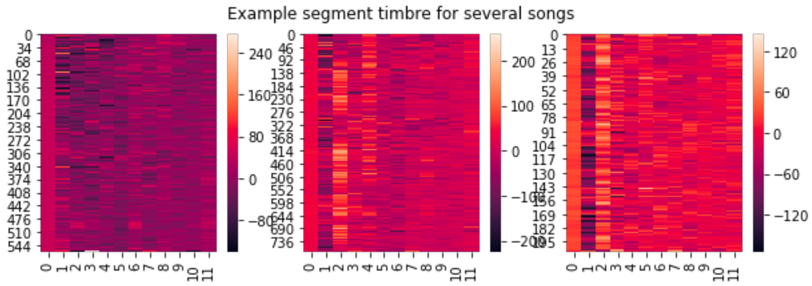
The number of segments per track is variable and each segment can itself be of variable length - typically they seem to be around 0.2 - 0.4 seconds but can be as long as 10 seconds or more.

Information on loudness at the start of each segment is shown below for three example songs. The outputs for loudness_max are similar.



The 12 dimensional arrays contained in the pitch and timbre columns are shown below for the same three example songs.





Genre and style information

We started to explore whether the MSD can be augmented with information from other sources. We tried to merge the data with genre and style information for Vienna University[10] by joining on the track IDs. Unfortunately, the results were incomplete, with more than half of records having no genre information and around three-quarters having no style information.

Taste Profile information

User dataset of MSD is https://labrosa.ee.columbia.edu/millionsong/tasteprofile which provides information on listens given by users to songs.

Additional step that needs to be taken is song->track mismatch clean up (https://labrosa.ee.columbia.edu/millionsong/blog/12-2-12-fixing-matching-errors)

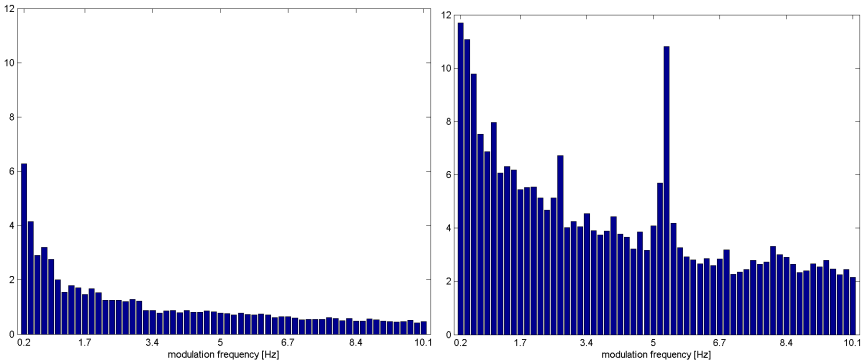
Audio information

http://www.ifs.tuwien.ac.at/mir/msd/download.html#groundtruth provides data on audio-extracted information in ARFF format (https://www.cs.waikato.ac.nz/~ml/weka/arff.html). We

[10] Available at http://www.ifs.tuwien.ac.at/mir/msd/download.html

started to look at rhythm histograms (http://www.ifs.tuwien.ac.at/mir/audiofeatureextraction.html#RH):

The Rhythm Histogram features are a descriptor for general rhythmics in an audio document. The magnitudes of each modulation frequency bin of all critical bands are summed up, to form a histogram of "rhythmic energy" per modulation frequency. The histogram contains 60 bins which reflect modulation frequency between 0 and 10 Hz. For a given piece of audio, the Rhythm Histogram feature set is calculated by taking the median of the histograms of every 6 second segment processed. E.g. classical vs. rock music:



ARFF format is difficult to read directly (version incompatibility) hence a direct CSV read is done. Match is done against track id. There are 60 measurements per track. The idea is to use these measurements as a feature to identify “hot” songs.

Next steps

The next steps are to better understand the features in the dataset and how they can be used for prediction, as well as to explore what other datasets could be linked to MSD in order to augment the features.