Novel Datasets for Evaluating Song Popularity Prediction Tasks

Michael Vötter, Maximilian Mayerl, Günther Specht, Eva Zangerle
Department of Computer Science, Universität Innsbruck, Austria
Email: {michael.voetter, maximilian.mayerl, guenther.specht, eva.zangerle}@uibk.ac.at

***Abstract*—Estimating the success of a song before its release is an important music industry task. Current work uses audio descriptors to predict the success (popularity) of a song, where typical measures of success are chart measures such as peak position and streaming measures such as listener-count. Currently, a wide range of datasets is used for that purpose, but most of them are not publicly available; likewise, available datasets are restricted either in size, available features, or popularity measures. This substantially impedes the evaluation of the predictive power of a wide range of models. Therefore, we present two novel datasets called HSP-S and HSP-L based on data from AcousticBrainz, Billboard Hot 100, the Million Song Dataset, and last.fm. Both datasets contain audio features, Mel-spectrograms as well as streaming listener- and play-counts. The larger HSP-L dataset contains 73,482 songs, whereas the smaller HSP-S dataset contains 7,736 songs and additionally features Billboard Hot 100 chart measures. In contrast to previous publicly available datasets, our datasets contain substantially more songs and richer and more diverse features. We solely utilize data from the public domain, allowing to evaluate and compare a wide range of models on our datasets. To demonstrate the use of the datasets, we perform a regression and a classification (popular/unpopular) task on both datasets using a wide variety of models to predict song popularity for all target measures.**

***Index Terms*—hit song prediction, popularity prediction, dataset, music information retrieval, audio features**

I. Introduction

An important task in the music industry is the prediction of hit songs, or more broadly, song popularity. In most cases, popularity is determined by sales, listening data on streaming platforms, chart positions, or the number of weeks a song stayed in the charts. In general, the goal is to predict the success of a song before or shortly after its release. This can, e.g., be of great interest to record labels to decide whether they should support a song. It can also be a tool for musicians to get feedback on a song during creation and tweak their song towards success. Also, music platforms may use such predictors to identify potentially popular tracks to recommend these to users with an inclination to popular content [1].

A number of approaches model the task as a regression task, relying on a continuous popularity value such as listener-counts or chart positions as their target [2]–[4], while others model it as a classification task based on classes, such as *hit* and *non-hit* as their target [5]–[7]. To this end, the two most common popularity measures are chart data and listening data collected by online music platforms such as last.fm and Spotify . Here, two common measures to quantify the popularity of a song are listener- and play-counts. Both of them are generated by counting unique listeners or song listening events on the particular platform. In contrast, charts are not limited to a particular platform. They usually include sales data and other measures of popularity in addition to streaming data. Details on the criteria used to determine the Billboard Hot 100 charts are discussed by Lao et al. [8].

Current approaches towards popularity prediction include neural network-based models such as the convolutional neural networks (CNN) based approach proposed in [3], [4], or more traditional approaches such as logistic regression [9], [10], decision trees [6] and SVM based models [7], [9], [11]. Typically, song popularity prediction approaches are evaluated in an offline evaluation setup. This requires sufficiently large datasets that provide a wide variety of song features and popularity values. However, there are no publicly available datasets that fulfill these criteria. In prior work [2], we released the currently only available dataset for popularity prediction, however, it is limited in the number of songs contained, the number of available features, and popularity scores.

In this paper, we present two novel datasets for hit song prediction (HSP): HSP-S and HSP-L. They are substantially larger than currently available dataset with respect to the number of features contained. Both datasets provide high- and low-level audio features stemming from AcousticBrainz [12] and short representative MP3 samples, as well as Mel-spectrogram features from the same samples. Further, we include listener- and play-counts gathered from last.fm for both datasets. Our larger dataset, HSP-L, contains 73,482 songs with audio features, listener- and play-counts, making it substantially larger than previous datasets. In addition, we provide the release year information for 65,575 songs as provided by the Million Song Dataset [13]. Therefore, it fits well for deep learning approaches. The smaller dataset, HSP-S, contains 7,736 songs, listener- and play-counts as well as Billboard Hot 100 data, and release year information is available for 7,449 songs. Hence, it is well suited to run experiments that require chart data. Both proposed datasets are published on Zenodo[1] and the source code of our evaluation is available on GitHub[2].

To show the utility of the datasets, we conducted experiments based on the two most widely used approaches regarding hit song prediction: (i) regression (e.g., performed by [2]–[4], [9], [10]) and (ii) classification (e.g., performed

[1]https://doi.org/10.5281/zenodo.5383858
[2]https://github.com/dbis-uibk/hit-prediction-code/tree/ism2021

978-1-6654-3734-9/21/$31.00 ©2021 IEEE
DOI 10.1109/ISM52913.2021.00034

TABLE I
DATASETS UTILIZED FOR INTRINSIC HIT SONG PREDICTION, WHERE HSP-S AND HSP-L ARE THE DATASETS PROPOSED IN THIS WORK. NOTATION: PD—DATASET STEMS FROM PUBLIC DOMAIN, AV—DATASET IS PUBLICLY AVAILABLE.

Paper	Type of audio features	Mel-Spect. Features	Charts Data	Listening Data	PD	AV	No. of Songs
[3]	audio	yes	-	KKBOX	no	no	approx. 125k
[4]	audio	yes	-	KKBOX	no	no	approx. 2M
[11]	in-house audio database	-	AU, UK, US	-	no	no	approx. 1,700
[6]	in-house audio database	-	UK	-	no	no	266
[14]	lyrics features	-	US	-	no	no	6,815
[15]	EchoNest features	-	UK	-	yes	no	5,947
[7], [16]	HiFind database of music	-	-	-	no	no	32,000
[2]	Essentia audio features	no	US	-	yes	yes	11,664
HSP-S	Essentia audio features	yes	US	last.fm	yes	yes	7,736
HSP-L	Essentia audio features	yes	-	last.fm	yes	yes	73,482

by [5], [11], [16]). We conducted experiments using linear models, random forests, support vector machines, feedforward neural networks, and convolutional neural networks.

The main contributions of this work are two novel datasets (HSP-S and HSP-L) that (i) provide substantially more songs, (ii) contain a richer and more diverse feature set compared to previous datasets that are mostly not publicly available (cf. Table I), and (iii) utilize solely data from the public domain.

These datasets can be used for a variety of popularity prediction approaches and evaluation setups. Particularly, we believe that for deep learning approaches, these comprehensive datasets can be a key asset. We demonstrate the utility of our datasets by a set of experiments and set baseline results for a broad spectrum of models. We believe that these datasets can enable further research on song popularity prediction as particularly the HSP-L dataset contains a large number of samples and features that can be leveraged by deep learning approaches. Both datasets provide a rich set of audio features describing these songs and can hence also be utilized for several further music retrieval and analysis tasks.

II. RELATED WORK

In the following, we discuss previous approaches towards song popularity prediction and present the used datasets.

Dhanaraj and Logan [11] aim to predict the success of a song based on acoustic and lyrics features. They investigate the usage of an SVM and boosting classifiers to automatically determine whether a song is a hit song and compare their models with a random baseline. They use an in-house dataset containing 18,500 songs which they used to compute the features. Frieler et al. [6] use melodic features as input for a random forest classifier that is trained to distinguish commercial successful pop songs from less successful pop songs. Their dataset consists of 266 pop songs which they categorized into a hit and a non-hit class using a k-means clustering using chart data from the UK. In contrast, Singhi and Brown [14] use lyric features to predict whether a song belongs to the hit or non-hit class. Their lyrics features include 31 rhyme, syllable, and meter features that they process with a Bayesian network. Their notion of a hit song is a song that made it to the Billboard Year-End Hot 100 singles charts between 2008-2013. For non-hits (flops) they choose songs from the same set of artists that did not make it to the charts. The dataset contains 492 hits and 6,323 flops. Ni et al. [15] aim to distinguish songs that made it to the top five of the UK charts (hits) from less popular songs that resided in the range 30-40 in these charts. They use musical features from EchoNest that they feed into a shifting perceptron. The dataset utilized contains 5,947 unique songs. Pachet and Roy [16] introduce the HiFind Database containing music metadata that are categorized into 16 categories covering a broad spectrum of a song such as style, genre, or instruments. Each of these categories contains more specific binary features that e.g., state if an acoustic guitar is present in the given song. The dataset contains some notion of popularity in the form of popularity labels (low, medium, and high). They predict those three labels using an SVM classifier with an RBF kernel. Both Yang et al. [3] and Yu et al. [4] use a neural network-based model trained on a dataset that contains popularity measures based on data from KKBOX Inc, a Taiwanese music streaming platform. The dataset used by [3] was collected between 10/2012 and 09/2013 and contains 125k songs with play-counts of 30k users. The dataset used by [4] was collected from 01/2016 to 06/2017 and contains play-counts of 30k users and 2M songs.

In contrast to the previously discussed literature, our previous work [2] provides a publicly available dataset. We combined data from the Million Song Dataset with chart data from the Billboard Hot 100 charts. In terms of features, we include Essentia audio features extracted from 30-second representative samples resulting in a dataset with 5,832 hit songs and the same number of non-hit songs. For the prediction, we propose a regression-based neural network model to predict the peak positions of a given song based on high- and low-level features for which we published the source code as well.

TABLE I provides an overview of datasets that have been used for hit song and popularity prediction. Only a single dataset is currently publicly available, which on the one hand, makes it hard to develop robust approaches, and on the other hand, to compare popularity prediction approaches. In contrast to most previous works, our datasets are based on publicly available data and are published. Furthermore, the HSP-L dataset is also substantially larger than most previous datasets.

III. DATASETS

For the tasks of hit song prediction (HSP) and song popularity prediction, we propose two datasets with complementary characteristics: The smaller dataset, *HSP-S*, contains streaming popularity measures gathered from last.fm and chart measures obtained from the Billboard Hot 100 charts. For this dataset, the number of tracks on the Billboard charts puts a natural cap on the number of tracks. Therefore, we also propose a second, larger dataset, *HSP-L*, that focuses on listener- and play-counts on streaming platforms as the target measure of success. Both datasets contain audio features stemming from AcousticBrainz as well as audio features we extract from short MP3 samples that are at least 30-seconds long (stemming from a private MP3 sample collection of the Million Song Dataset). Additionally, both datasets contain Mel-spectrograms computed on the same MP3 samples and release year information from the Million Song Dataset for most of the songs.

An overview of the features contained in the two datasets is given in TABLE II. In the following, we describe the creation of the two datasets (Sections III-A and III-B) and further detail the audio features (Section III-C) and success measures provided (Section III-D).

TABLE II
FEATURES AVAILABLE ON THE DATASETS (CLASSIFICATION TAKEN FROM [2]).

Feature	HSP-S	HSP-L
Listener-count	✓	✓
Play-count	✓	✓
Peak position	✓	
Weeks in charts	✓	
Low-level descriptors	✓	✓
Mood	✓	✓
Voice	✓	✓
Rhythm/Beat	✓	✓
Chords	✓	✓
Mel-spectrograms	✓	✓
Number of songs with release year	7,449	65,575
Total number of songs	7,736	73,482

A. HSP-S Dataset

The HSP-S dataset is the smaller of both datasets. For its creation, we match the Billboard Hot 100 charts between 1958-08-11 and 2019-07-06 with the Million Song Dataset (MSD) [13] based on artist name and song title using the following procedure.

First, we convert artist and title strings in both source datasets to lower case and strip white space at the beginning and end of each string. After that, we remove all characters that are not alphanumerical or white space, which e.g., removes differences in punctuation such as "ft." vs. "ft" for featuring. The last step is to substitute multiple successive white space characters with a single blank space character. Further, we ensure that there are no entries with an empty artist or title.

Before matching both source datasets (Billboard Hot 100 and the MSD), we drop artist-title duplicates in each dataset to ensure to not include the same song multiple times. Based on the resulting strings, we compute the overlapping tracks based on artist name and song title. This results in 14,248 songs contained in both the MSD and the Billboard Hot 100.

To include last.fm listener- and play-count data, we utilize the MBID (MusicBrainz IDentifier) mapping[3] between last.fm and AcousticBrainz. We performed the previously described artist and title cleaning and dropped entries where either of them is empty. This leaves us with a set of hit songs contained in the MSD. To ensure a balanced dataset with respect to hits and non-hits, which is important for binary classification, we randomly sampled one non-hit from MSD per hit. After that, we assigned a universally unique identifier (UUID) to each unique title-artist pair in the set of songs that results from merging the Billboard, MSD, last.fm, and AcousticBrainz data. Further, we join those UUIDs into one if two songs have the same MSD_ID, EchoNest ID, or MBID, resulting in a unique ID that can be used to identify unique songs in our dataset. In a final step, we merged the extracted features (as described in Section III-C) with the popularity measure values. This results in a dataset containing different high- and low-level audio features (audio and Mel-spectrogram features) from AcousticBrainz and short MP3 samples (see Section III-C). It contains 7,736 songs in total with last.fm listener- and play-count data for all songs and the peak position and the weeks in the Billboard Hot 100 charts for all hits.

For collecting release year information, we utilize the release year mapping with MSD_IDs provided for the Million Song Dataset[4]. To compute the final mapping between the UUIDs and the release year, we apply the following procedure. First, we ensure that each original UUID (before the merging step) maps to the same release year. If this is not the case, we drop the affected UUID. In the next step, we merge the data records using the MSD_ID. This results in an unambiguous mapping between UUIDs and release year for 7,449 of the 7,736 songs (96.30% of all songs).

B. HSP-L Dataset

The *HSP-L* dataset is substantially larger than the HSP-S dataset as it is not tied to charts data, where only a limited amount of songs is available. Instead, HSP-L focuses on last.fm listener- and play-counts as target measures.

Similar to the HSP-S dataset, we use the mapping of the Million Song Dataset to AcousticBrainz as a starting point and matched all entries with last.fm popularity data (listener- and play-count; see Section III-D). Additionally, we merge all the available features resulting from the application of our feature selection algorithm described in Section III-C. This results in a dataset containing the same set of audio features that are also present in the HSP-S dataset (see Section III-A). We include the release year following the same procedure as used for the HSP-S dataset. Please note that the UUID is assigned to songs individually for each dataset and hence, does not

[3]https://musicbrainz.org/doc/MusicBrainz_Identifier

[4]http://millionsongdataset.com/sites/default/files/AdditionalFiles/tracks_per_year.txt

refer to the same songs. Therefore, it is necessary to compute a separate mapping of UUIDs and release year for the HSP-L dataset. In contrast to the HSP-S dataset, this dataset only contains popularity data stemming from last.fm and hence, does not suffer from missing popularity values for songs that never made it to the Billboard Hot 100 charts. Therefore, this dataset does not contain a hit and non-hit class and hence, we did not balance the dataset. Further, its size is not limited by the number of songs that appeared in the charts. The final dataset contains 73,482 songs with release year information for 65,575 songs (89.24% of all songs).

C. Audio Features

Our datasets contain audio features extracted with well-known libraries: we use LibRosa [17] to extract the Mel-spectrograms in dB-scale with 128 bins and 1200 slots along the time axis and Essentia [18] for further audio features. We include the Essentia low-level features consisting of acoustic descriptors characterizing the song using average loudness, bark bands, erb bands, mel bands, to name some of the dynamics and spectral features extracted for a song. Further, rhythm and tonal features are included[5]. The included Essentia high-level features are inferred using the pre-trained classifiers bundled with the framework and include danceability, mood, genre, and vocal features.

AcousticBrainz may provide multiple versions for the same song. Further, the Million Song Dataset contains duplicates. In either case, we only keep one version per song, that we identify using the assigned UUID as follows. In a first step, we validate the quality of our features before selecting the most representative version. Thus, we compare the histograms of each feature for both sources (AcousticBrainz and the MP3 samples). We compute a histogram for each feature per dataset. This step reveals that for multiple features, the histograms contain outliers: Some of these histograms show a Gaussian distribution for the predicted probabilities of a given high-level Essentia feature with a significant outlier (spike) for a small value range. Those spikes are caused by specific beta software versions of the low-level Essentia extractors used by AcousticBrainz for some of the song's feature versions. We suspect that those spikes occur because these extractors compute certain values too often. Therefore, we first remove samples that were created with different builds of the Essentia v2.1 beta1 software version that seemed to be affected. This substantially reduces the peaks. Removing features created with those software versions ensures that the remaining set of features are of high quality. The resulting set still contains multiple feature versions of the same song (identified by the assigned UUID) as this cleaning step does not remove all duplicate versions of a song. To subsequently select the most representative version for each song, we select the feature version closest to the mean of all remaining feature sets of the same song (i.e., we use the sample closest to the centroid of each song's cluster). Along the lines of previous research [2], we use cosine distance based on the set of all audio features to determine the song version most similar to the centroid (the average). This leaves us with a single high- and low-level feature version per unique song (UUID). Subsequently, we combine the pairs of high- and low-level features of each unique song. After applying these cleaning, selection, and join steps for features stemming from AcousticBrainz and shorter audio samples from the Million Song Dataset (MSD) [19], we merge the resulting features from both sources and only keep those songs where features from both are available, leaving us with a single version of these features for each song. Further, we also compute and include Mel-spectrograms on the shorter audio samples. We dropped a song in the final dataset in case we could not compute the Mel-spectrogram features. This leaves us with a dataset containing a single version of all previously mentioned features for each unique song.

[5]These features are further described in Essentia's documentation: https://essentia.upf.edu/streaming_extractor_music.html.

D. Popularity Measures

To obtain popularity measures that can be used as target value for the task of song popularity prediction, we use Billboard chart information (as previously done by [2], [14]) as well as listener-counts and play-counts (as previously employed by [3], [4]) from last.fm. In some rare cases, there are duplicate entries in either last.fm data or Billboard charts for songs. To eliminate those, we use the best score (the highest score for listener-, play-counts, and weeks in charts; the lowest score for peak ranking in charts) as we consider this the most representative score. Besides charts and listener- and play-count data, we also compute Yang et al.'s hit-score [3]. This score allows mitigating scenarios in which the play-count of a song is driven by only a small number of users by multiplying the log of the listener-count with the log of the play-count.

TABLE III
PLAY- AND LISTENER-COUNT STATISTICS ON THE DATASETS (PC = PLAY-COUNT, LC = LISTENER-COUNT).

Property	HSP-S	HSP-L
Minimum LC	0	0
Median LC	17,210	8,797
Maximum LC	2,119,960	2,119,960
Minimum PC	0	0
Median PC	53,824	27,554
Maximum PC	22,660,386	22,660,386

TABLE III shows further details on listener- and play-counts data in both datasets. We observe that the range of each popularity measure is comparable among both datasets. Further, it can be seen that the mean for streaming popularity measures substantially differs. This is caused by the different sampling strategies used for each dataset. While the number of 7,736 songs in the HSP-S dataset is smaller than the number of songs contained in the dataset published in [2], it contains more features and more popularity measures (also including charts data) that can be utilized for experiments. To the best of our knowledge, the HSP-L dataset with 73,482 songs substantially exceeds the size of current, publicly available

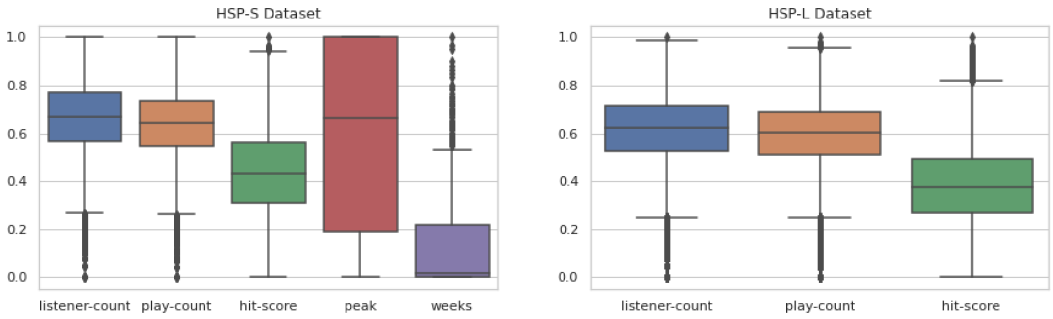

Fig. 1. Target label distribution for the HSP-S and HSP-L datasets (normalized; listener- and play-count in log scale).

datasets for the task of song popularity prediction. It is by far the largest dataset containing a wide variety of features and streaming popularity measures, that is publicly available.

Fig. 1 shows the distribution of the target labels in our datasets. For both the listener- and the play-count we compute the log before scaling it to the range 0 to 1. We note that peak is the only target where lower values means that the song is more popular, while for all other targets a higher value means higher popularity. We observe that naturally, the charts-based targets (peak and weeks) differ from the remaining targets. Further, Fig. 1 shows that the median, as well as Q3, and the maximum of the HSP-L targets are lower in comparison to their HSP-S counterpart. This indicates that the HSP-S dataset on average contains more popular songs, which is likely caused by the fact, that half of them made it to the charts. For such chart hits it can be assumed that they are examples of very popular songs.

IV. Experiments

In this section, we present the experiments that we conduct to showcase the utility and applicability of both datasets presented in Section III. Our experiments cover the two main approaches towards music popularity prediction (e.g., in [2]–[5], [11]): (i) modeling popularity prediction as a regression task, and (ii) modeling it as a classification task and provide baseline results for a variety of models.

A. Evaluated Tasks

For both the regression and classification task, a wide range of sources and representations of popularity measures were used in previous works. For instance, Yu et al. [4] use play-counts as a popularity measure tackling a regression task while in our previous work [2], we use the peak position in charts. Similarly, there is a difference in how popularity is defined in classification tasks. While Pachet [7] uses three popularity classes, Ni et al. [15] use a hit and a non-hit class where hits are songs that appeared in the top five of the charts and non-hit are songs that appeared in the range between 30 and 40.

To cover most of these representations used in previous works, we use all four representations contained in our datasets (peak position and weeks in charts; listener- and play-counts) and the hit-score defined by Yang et al. [3] in our experiments. We conduct *regression task* experiments for each popularity measure along the lines of [2], [3]; we utilize four different models that are further described in Section IV-B to predict the respective popularity measure based on features stemming from AcousticBrainz and two models to compute predictions based on the Mel-spectrograms. For the experiments conducted on the *classification task*, we use the same popularity measures as for the regression task mapped to a two-class classification problem as detailed in Section IV-C. We conduct experiments with four models using the Essentia audio features stemming from AcousticBrainz and two models using the Mel-spectrograms. Due to space constraints, we do not provide the experiments based on Essentia features computed from the short audio samples as they do not cover the full length of a song in contrast to those stemming from AcousticBrainz.

B. Models

Along the lines of previous popularity prediction approaches [2], [3], [5], [7], [11], we propose the following models for the conducted regression and classification experiments. We use six models to compute predictions for the regression task: (i) a linear regression model, (ii) a random forest model (iii) an SVM-based model, and (iv) a deep feedforward neural network model used in [2] that all rely on AcusticBrainz features, (v) a linear regression-based model using the mean and standard deviation along the time axis of the Mel-spectrograms based on the m1 model in [3], and (vi) a CNN model using Mel-spectrogram features. For the classification task, we adapt the models of the regression task to perform classification. Instead of the linear regression model, we use a logistic regression classifier as a representative of linear models as used in [9], [10].

We use the implementations of scikit-learn [20] for all non-network models. The SVM utilizes the default RBF kernel; we also keep all other default parameters for all models. We use the SVR implementation for the regression task and the SVC implementation for the classification task. To use the SVC implementation with our two-class definition of popularity, we need to convert the two-class (popular/unpopular) popularity measure (represented as a one-hot vector) to a single label

value to train the model. This is done by using the index of the respective class. To obtain predictions, we reverse this transformation by creating a one-hot encoded vector where we set the respective class to one.

The Wide and Deep neural network model is based on the model proposed in [2] and uses the implementation we provide (based on the Keras API of TensorFlow[6]). The model consists of an input layer that is split into two types of input: high-level Essentia audio features that are directly fed into the wide part of the network and low-level Essentia features that are consumed by the deep part of the network. Each multi-dimensional low-level feature gets mapped to its feature aggregation block that consists of a dense layer with an input size fitting the dimension of the features. These features are aggregated to a single-valued higher-level feature representation using ELU to activate the output of each block. The resulting feature representations are then combined with the high-level Essentia audio features in a concatenation layer. This layer is the input of a dense layer block of the neural network and contains one neuron per feature. This block contains two units, each consisting of a dense layer with ELU activation followed by a dropout layer (0.1 dropout rate). Each unit has the same input and output size. The final output layer of the network consists of a single neuron without an output activation to compute predictions for the regression task.

Our CNN model is based on the fully convolutional neural network (FCN) by Choi et al. [21]. We use a Keras port of the implementation provided by Won et al. [22] (written in PyTorch)[7]. It consists of five blocks, where each block consists of a 2D convolutional layer, followed by a batch normalization layer, and a 2D max pooling layer. The first and last block have 64 filters while all others have 128. All convolution kernel sizes are 3x3 with stride 1. The size of the first three max pooling layers is 2x4 and the last two have a size of 3x5 and 4x4 respectively and each block uses ReLU as its activation function. The input size 128x1200 is zero padded to get an input size of 128x1296. This ensures, that the five blocks reduces the size to 1x1 with 64 filters. After theses blocks we apply a dropout layer with 0.5 dropout probability and use a dense layer to get the appropriate output size (1 for regression task; 2 for classification task).

C. Experimental Setup

The main purpose of these experiments is to show the usability and applicability of our proposed datasets. To evaluate our experiments, we use 5-fold cross validation. In terms of features, we use all Essentia features listed in [2], in addition to Mel-spectrogram features in dB-scale extracted using LibRosa [17]. The Essentia features are scaled to the range $[0, 1]$ by applying scikit-learn's min-max scaler.

The models implemented based on scikit-learn are trained using the default settings. For non-network models, we compute a single result per fold that we average over all five folds to get the reported scores. We consider the number of epochs a hyper-parameter as done by [2] and train the neural network-based models using a grid search where we evaluate every N epochs up to a maximum number of epochs and report the results for the best epoch hyper-parameter configuration. We train the Wide and Deep model with a maximum of 2,500 epochs and evaluate the performance every 100 epochs. For the FCN we use a maximum of 50 epochs and evaluate every 10 epochs to include the range of the best epochs reported for this kind of model by Choi et al. [21]. The networks were trained using the Adam optimizer with the mean squared error loss function for the regression task.

As for the popularity measures, we use zero as the popularity value for non-hits along the lines of [2] and use 101 as the peak position popularity value for non-hit songs.

For the binary classification experiments, we need to map the discrete popularity values to classes. Along the lines of [5], [6], [11], [14], we use two categories *popular (hit)* and *unpopular (non-hit)*. Consequently, we need to define a criterion to distinguish these two classes. For charts data, we use whether a song appeared in the charts (hits) or not (non-hits) as the criterion, providing us with a balanced number of hits and non-hits as the HSP-S dataset is balanced. When using last.fm data, we use the median of the respective popularity value as a decision boundary similar to [5] to ensure a balanced dataset. Note that this results in a different definition for popular and unpopular songs depending on the underlying popularity measure. This needs to be taken into account when comparing results based on charts, listener-counts, play-counts, and the hit-score defined by Yang et al. [3].

For the evaluation of the regression task, we compute Spearman's and Kendall's ranking correlation coefficients to capture how well the models rank songs. We argue that the underlying level of measurement used for chart popularity measures cannot be considered an interval scale because all non-hit songs share a common popularity value (O weeks or peak position 101). This prevents computing a distance between two non-hits. Arguably, not all non-hits are equal with respect to the popularity prediction task as they are not equally popular. This is also indicated by different popularity values in terms of listener- or play-counts seen among non-hit songs. Therefore, the ranking coefficient scores can be assumed to be more expressive than, e.g., the mean absolute error or the root mean squared error because all underlying levels of measurement fulfill the properties of an ordinal scale.

To evaluate the classification task, we report the macro-averaged accuracy and F1. Note that we model popular and unpopular as two separate classes instead of one binary label. We chose macro averaging to compensate for the slight imbalance between the popular and unpopular class resulting from the used median to distinguish the two classes because there is a small number of songs whose popularity values are equal to the median value and therefore need to be assigned to one class. Notice that this only happens for the listener-count, play-count, and hit-score experiments.

[6]https://www.tensorflow.org/

[7]https://github.com/minzwon/sota-music-tagging-models

TABLE IV
EVALUATION RESULTS FOR THE REGRESSION TASK (LC = LISTENER-COUNT, PC = PLAY-COUNT, HS = HIT-SCORE); BEST RESULT PER DATASET, TARGET, AND METRIC IN BOLD.

		Spearman's ρ						Kendall's τ					
Target	**Dataset**	**LinReg**	**RndForest**	**SVM**	**W&D**	**MelSp-LinReg**	**FCN**	**LinReg**	**RndForest**	**SVM**	**W&D**	**MelSp-LinReg**	**FCN**
Peak	HSP-S	**0.468**	0.426	0.393	0.454	0.308	0.398	**0.343**	0.311	0.283	0.332	0.224	0.289
Weeks	HSP-S	**0.420**	0.349	0.397	0.417	0.277	0.362	**0.313**	0.260	0.287	0.312	0.204	0.266
LC	HSP-S	0.352	0.304	0.269	**0.368**	0.322	0.333	0.241	0.207	0.182	**0.252**	0.218	0.227
LC	HSP-L	0.334	0.268	0.204	**0.342**	0.272	-	0.226	0.181	0.136	**0.232**	0.184	-
PC	HSP-S	0.342	0.315	0.284	**0.362**	0.320	0.344	0.234	0.215	0.192	**0.247**	0.217	0.234
PC	HSP-L	0.361	0.286	0.231	**0.370**	0.291	-	0.241	0.193	0.155	**0.252**	0.197	-
HS	HSP-S	0.433	**0.447**	0.352	0.440	0.365	0.376	0.298	**0.308**	0.239	0.302	0.248	0.256
HS	HSP-L	0.403	0.416	0.346	**0.430**	0.320	-	0.274	0.284	0.233	**0.294**	0.217	-

V. RESULTS AND DISCUSSION

TABLE IV presents the results of all models on both proposed datasets for the regression task. For charts data predictions, linear regression provides the best results, whereas for predicting popularity in terms of streaming counts, the Wide and Deep network provides the best results (except for the hit score prediction for the HSP-S dataset).

An interesting finding is, that the linear regression model outperforms all other models on charts-based target measures while the Wide and Deep model shows a good predictive performance for the remaining targets. Similarly, the neural network-based FCN model outperforms the simpler linear regression-based model (MelSp-LinReg). Furthermore, comparing the results of the individual models for both datasets, we observe that the linear regression model and the Wide and Deep model perform better on the larger dataset when predicting the play-counts as target. Here, Spearman's ρ (and Kendall's τ) increase from 0.342 (0.234) to 0.361 (0.241) and 0.362 (0.247) to 0.370 (0.252), respectively. We lead this back to the fact that both datasets are different in terms of the distribution of the respective target labels as depicted in Fig. 1.

Notably, models consuming the Mel-spectrograms do not outperform the models using the Essentia features. To the best of our knowledge, this is a novel finding but it should be noted, that we were not able to gather results for the FCN model on the larger HSP-L dataset as the current implementation requires more than 256GB of memory during training. In future work, empowered by our datasets, it will be interesting to see how models consuming Mel-spectrograms compare to more traditional approaches. Another interesting observation is, that except for the random forest model, all models achieved the highest correlation scores for the peak positions popularity measure, followed by the hit-score popularity measure (looking at the HSP-S) dataset.

The results of the binary classification task are shown in TABLE V. Analogously to the regression task results, all models reveal a better predictive performance on the smaller HSP-S dataset. In the HSP-S dataset, 50% of the songs made it to the Billboard Hot 100 charts. This leads to a different label distribution as seen in Fig. 1, where the target measures listener-count, play-count and hit-score in the HSP-S dataset have a higher average and a larger interquartile-range. Further, we observe, that the SVM model achieves the highest scores with an accuracy of 0.715 and an F1 score 0.714 on the chart-based classification, followed by the logistic regression model with an accuracy of 0.713 and an F1 score of 0.712. The two experiments based on the classes derived from peak position and weeks in charts reveal the same accuracy in case of models that do not use randomness and similar accuracy in case of models that do use some kind of randomness. Hence, we only state these results once (cf. the row"Charts"). These results can be lead back to the fact that the hit vs. non-hit definition leads to the same classes for both peak position and weeks. The only difference between the two representations is the meaning of both vector entries. If the first entry is one, it is a non-hit when using weeks in charts and a hit when using the peak position. Again, it can be seen that the Wide and Deep model achieves the best results on streaming based targets except for listener-count prediction on the HSP-S dataset, where it achieves the second best results with and accuracy of 0.647.

In contrast to the regression based results, the neural network based model is not able to outperform the simpler logistic regression model that uses the mean and standard deviation of the Mel-spectrograms as input (MelSp-LogReg). Again, we were not able to gather results for the FCN model on the HSP-L dataset as the model and dataset due to memory constraints (256 GB of memory did not suffice).

To conclude, we see that the two datasets allow getting valuable insights on the performance of various models to tackle different kinds of popularity prediction tasks using different kinds of features. Having both, Essentia features and Mel-spectrogram features available for the same dataset enables future work that compares models using either of them. However, we still see a few limitations of our approach, which we will elaborate on in the following. We acknowledge that the presented results are biased towards western and mainly commercial music due to the data sources utilized. Further, the used definitions of success (Billboard charts and last.fm streaming measures) introduce a platform bias.

TABLE V
EVALUATION RESULTS FOR THE BINARY CLASSIFICATION TASK (LC = LISTENER-COUNT, PC = PLAY-COUNT, HS = HIT-SCORE); BEST RESULT PER DATASET, TARGET, AND METRIC IN BOLD.

		Accuracy						F1					
Target	**Dataset**	**LogReg**	**RndForest**	**SVM**	**W&D**	**MelSp-LogReg**	**FCN**	**LogReg**	**RndForest**	**SVM**	**W&D**	**MelSp-LogReg**	**FCN**
Charts	HSP-S	0.713	0.707	**0.715**	0.711	0.638	0.649	0.712	0.707	**0.714**	0.710	0.638	0.644
LC	HSP-S	0.641	**0.648**	0.646	0.647	0.627	0.600	0.641	**0.648**	0.644	0.646	0.627	0.597
LC	HSP-L	0.623	0.635	0.631	**0.644**	0.606	-	0.623	0.635	0.630	**0.644**	0.605	-
PC	HSP-S	0.655	0.653	0.650	**0.658**	0.634	0.601	0.655	0.652	0.648	**0.658**	0.634	0.598
PC	HSP-L	0.637	0.648	0.646	**0.655**	0.617	-	0.637	0.648	0.645	**0.655**	0.617	-
HS	HSP-S	0.650	0.653	0.651	**0.655**	0.632	0.599	0.650	0.652	0.647	**0.654**	0.631	0.593
HS	HSP-L	0.630	0.644	0.640	**0.649**	0.613	-	0.629	0.644	0.639	**0.649**	0.612	-

VI. CONCLUSION

We present two datasets (HSP-S and HSP-L) that provide Essentia's high- and low-level audio features, and Mel-spectrograms. As for popularity measures, we include listener- and play-counts gathered from last.fm for both datasets, which also allows computing Yang et al's hit-score [3]. Furthermore, HSP-S also provides charts data extracted from the Billboard Hot 100. Our experiments show that both datasets allow comparing different types of models on different popularity prediction tasks. One crucial contribution is that our HSP-L dataset allows to train models that require a large number of training samples to increase their performance. We believe that our datasets will be used for investigating and evaluating a variety of music popularity prediction approaches; particularly given that deep neural networks may benefit from large numbers of training samples. In addition, it is now possible to run multiple models using the same dataset utilizing different sets and types of features stemming from multiple sources to compare the results using the same songs. Further, such experiments can take advantage of the different popularity measures. This allows, for instance, investigating differences among those measures and the approaches utilized.

REFERENCES

[1] D. Kowald, P. Muellner, E. Zangerle, C. Bauer, M. Schedl, and E. Lex, "Support the underground: characteristics of beyond-mainstream music listeners," *EPJ Data Science*, vol. 10, no. 1, p. 14, Mar 2021.

[2] E. Zangerle, M. Vötter, R. Huber, and Y.-H. Yang, "Hit song prediction: Leveraging low- and high-level audio features," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 319–326.

[3] L.-C. Yang, S.-Y. Chou, J.-Y. Liu, Y.-H. Yang, and Y.-A. Chen, "Revisiting the problem of audio-based hit song prediction using convolutional neural networks," in *Proceedings of the IEEE International Conference Acoustics, Speech and Signal Processing*, 2017, pp. 621–625.

[4] L.-C. Yu, Y.-H. Yang, Y.-N. Hung, and Y.-A. Chen, "Hit song prediction for pop music by siamese cnn with ranking loss," *arXiv preprint arXiv:1710.10814*, 2017.

[5] J. Lee and J. Lee, "Music Popularity: Metrics, Characteristics, and Audio-Based Prediction," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3173–3182, 2018.

[6] K. Frieler, K. Jakubowski, and D. Müllensiefen, "Is it the song and not the singer? Hit song prediction using structural features of melodies," *Yearbook of Music Psychology*, pp. 41–54, 2015.

[7] F. Pachet, "Hit Song Science," in *Music Data Mining*, 1st ed. Chapman & Hall/CRC Press Boca Raton, FL, 2012, pp. 305–326.

[8] J. Lao and K. H. Nguyen, "One-hit wonder or superstardom? the role of technology format on billboard's hot 100 performance," *online*, 2016. [Online]. Available: https://web.stanford.edu/~xhnguyen/BillboardandTechnology.pdf

[9] D. Herremans, D. Martens, and K. Sörensen, "Dance hit song prediction," *Journal of New Music Research*, vol. 43, no. 3, pp. 291–302, 2014.

[10] J. Fan and M. Casey, "Study of chinese and uk hit songs prediction," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, 2013, pp. 640–652.

[11] R. Dhanaraj and B. Logan, "Automatic prediction of hit songs." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2005.

[12] A. Porter, D. Bogdanov, R. Kaye, R. Tsukanov, and X. Serra, "Acousticbrainz: A community platform for gathering music information obtained from audio," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, vol. 787, 2015.

[13] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[14] A. Singhi and D. G. Brown, "Hit song detection using lyric features alone," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[15] Y. Ni, R. Santos-Rodriguez, M. Mcvicar, and T. De Bie, "Hit song science once again a science?" in *Proceedings of the International Workshop on Machine Learning and Music*, 2011, pp. 2–3.

[16] F. Pachet and P. Roy, "Hit song science is not yet a science." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2008, pp. 355–360.

[17] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[18] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepat, J. Salamon, J. R. Zapata González, X. Serra *et al.*, "Essentia: An audio analysis library for music information retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2013, pp. 493–498.

[19] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[21] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *arXiv:1606.00298*, Jun. 2016, arXiv: 1606.00298. [Online]. Available: http://arxiv.org/abs/1606.00298

[22] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of CNN-based Automatic Music Tagging Models," *arXiv:2006.00751*, Jun. 2020, arXiv: 2006.00751. [Online]. Available: http://arxiv.org/abs/2006.00751