Machine Listening and Music Production: A Discrimination-Generation Feedback Loop Using Amper Music and Icelandic Indie Audio Data

Alex Sigman
Amper Music
NYCDSA Capstone Project

Abstract

In this study, music descriptor classification models were trained and tested on music data generated via Amper Music's system. These models, as well as tempo/key induction and KMeans-based Laplacian form segmentation algorithms were applied to a compilation of Icelandic indie tracks that for the most part represented electronica and indie rock genres currently supported by Amper. In turn, the resulting analyses for each input track were employed to generate single-region and multi-region ("stitched") renders corresponding to that track. A given region was assigned one of the most frequently occurring descriptor classifications. In the case of multi-region renders, the region-lengths corresponded to the durations of induced form segments. Subsequently, descriptor classification models were applied to single-render and multi-region renders. While the classification results of the former were evaluated for accuracy, those of the latter were compared with the input track classifications to determine consistency/stability.

INTRODUCTION

Amper Music

Amper Music is an AI-driven music software company specializing in the generation of flexible royalty-free stock music for content creators (e.g., filmmakers, game designers, podcast producers, etc.). All of Amper's instrumental samples are professionally performed, recorded, and produced. Music categories are defined in terms of *descriptors.* A descriptor consists of a genre, subgenre, mood, and sentiment (negative, neutral, positive), and is defined in terms of the musical, instrumental, and audio (digital signal processing, or DSP) constraints (represented as tags) to which it is mapped. A team of asset creators continuously researches into trending musical styles and generates data conforming to the stylistic norms of the given descriptor. Currently, Amper supports seven unique genres, ranging from folk to hip-hop, each containing between one and four subgenres.

Users may engage with the Amper ecosystem through two means: *Score,* a web-based app, and the public API. Within the Score app, the user is asked to specify a duration for a given render (or may upload a video or audio file from which the duration may be extracted), select a desired genre, subgenre, and mood, and to choose from a list of "bands" (groups of instruments), which may be previewed prior to rendering. In addition, the user may adjust the lengths of the introduction and ending, and location of the climax within the segment. In seconds, a render complying with user specifications is generated via the *Composer,* a generative (rule-based) algorithm engine, and *Inferno*, the audio rendering engine.

Instrumentation, tempo, key, structure (intro/climax/outro placement) mood may be adjusted, enabling the user to iteratively generate and audition variations, while remaining within viable descriptor boundaries. Preferred renders may be downloaded and archived.

Machine Listening: Use-Cases and Motivations

Although Amper's music generation system does employs symbolic, as opposed to machine learning techniques (for a variety of reasons), and is fueled by human-generated data--an expert system, so to speak--there have recently arisen use-cases that would best be addressed through applying machine learning--or more accurately, machine *listening*--models.

One objective is to provide the asset creation team with a classification toolkit that would facilitate descriptor research (e.g., confirming tempo, tonic, formal analysis, and other musical characteristics), A/B testing, QC, measuring relative descriptor "distances," and so on. As the scale and complexity of the database increases (as well as the demand for new descriptors), such a time-saving toolkit will become of greater necessity.

Another pertains to end-user interaction. There have been a few requests amongst current users to enable render generation without needing to specify instrumentation, click through the genre-subgenre-mood tree, and to have a more convenient method of exploring the Amper descriptor space. As such, various efforts are underway to introduce more flexibility to the Score input model. A longstanding (and longer-term) goal has been to enable the user to upload a reference track, and for the system to return a render similar in nature with respect to style, structure, and instrumentation.

Project Methodology: A High-Level View

Achieving the implementation of the reference track paradigm would require not only the integration of source separation, instrument recognition, style classification, tempo/key induction, and form segmentation, but also determining reasonable criteria for input/output similarity. As shall become evident throughout this paper, as a) the ultimate concern is generation (as opposed to discrimination), b) the Amper system is trained to produce style/use-case specific music adhering to a unique (and evolving) set of quality standards, and c) there is in general a broad range of "acceptable" musical output, given an input (as compared to the domains of machine translation, recommender systems, or even image generation, for instance), and deriving such criteria is not a straightforward process.

As is indicated in the diagram below, the project workflow consisted of the following stages:

1) Training data was generated via the Amper render server, based upon a dictionary of active descriptors.

2) Data acoustic feature extraction was performed.

3) Descriptor classification models were trained on the data.

4) The trained models were applied to the external dataset (in this case, an Icelandic indie compilation), after removal of vocal tracks (Amper renders contain neither melodic/vocal lines nor lyrics).

5) Tempo and key induction models, as well as an (unsupervised) KMeans-based form segmentation model were run on the Icelandic indie tracks.

6) The output from these models was merged and binned.

7) The merged model predictions were used to generate:

a) **single region renders** (i.e., renders of the same duration as reference tracks, each assigned one of the most frequently occurring descriptors, tempi, and keys output by the classification models);

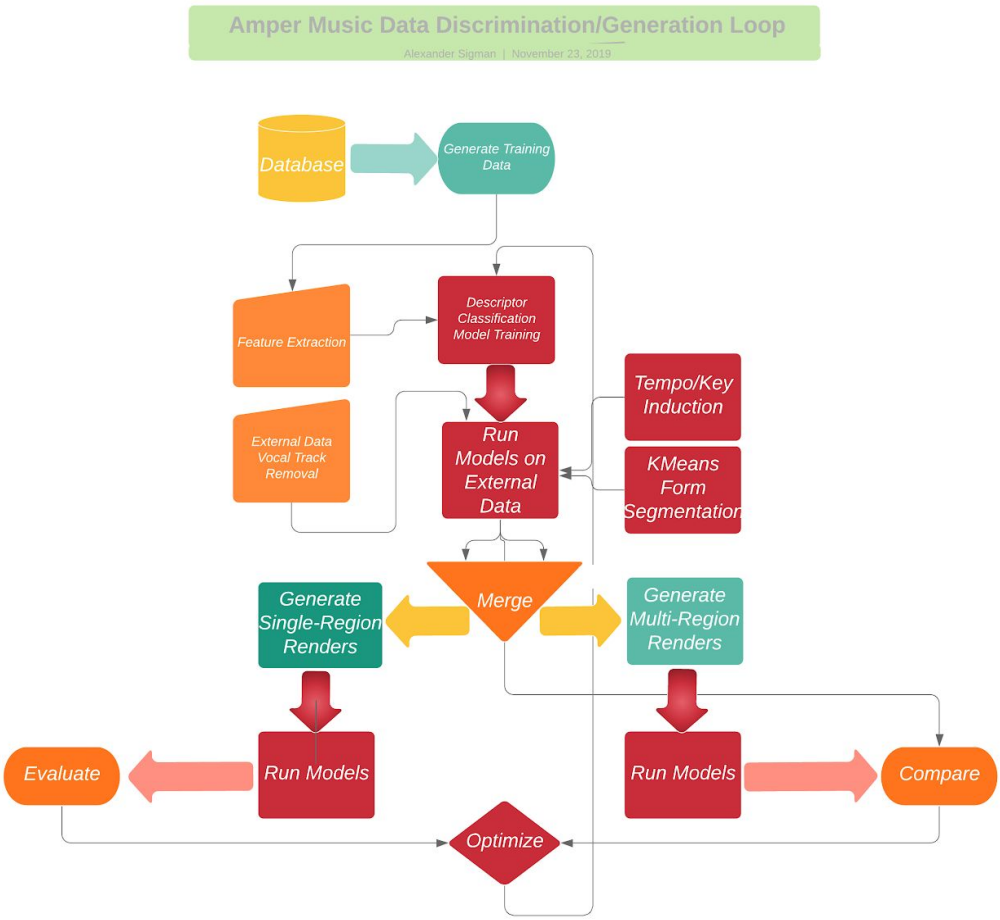
b) **multi-region renders** (i.e., renders of the same duration as reference tracks, each divided into multiple regions, whose lengths correspond to those of the input tracks"s induced formal segments. Each region is assigned the most frequently occurring descriptor, tempo, and keys output by the classification models *for the corresponding formal segment in the input track*)

8) In turn, classifiers were applied to both the single region and multi-region renders (a "dog-fooding" procedure).

9) Classification results for single-region renders were evaluated for accuracy (as a sanity check/validation of model performance).

10) Classification results for multi-region renders were compared to those of the original input tracks.

11) Insights obtained from steps (10) and (11) were used to inform subsequent data collection, model selection, and hyperparameter tweaking.



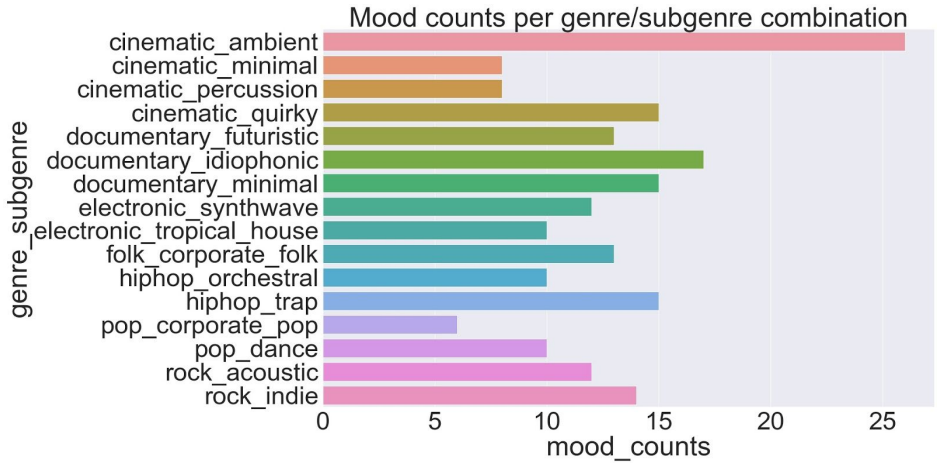
EDA: Amper Training Data

In the interest of creating a dataset that was balanced amongst classes, and heterogeneous within each class, 33 renders of each active descriptor were generated at one of three unique durations: 10, 30, and 60 seconds, respectively, thus yielding 99 examples in total for each of the 204 active descriptors. In the interest of model robustness, a random tempo and instrumentation was selected for each render.[1] In total, 20,196 examples were generated in the most recent batch.[2]

Each descriptor is identified by an internal name key, and assigned values for genre, subgenre, mood, sentiment (negative, positive, or neutral), and a unique ID. E,g.: "cinematic_percussion_primal_pulsing": ["powerful", "neutral", 46, "cinematic", "percussion"]

[1] These random choices are constrained to the range of allowed tempi (20-200 beats per minute [BPM]), and the set of active instruments associated with the given descriptor, respectively.
[2] Due to the periodic introduction of new descriptors, and deprecation of older ones, the data collection process must be repeated on a regular basis to ensure that the models are updated.

The graph below indicates the distribution of moods per genre/subgenre combination. It is evident that moods are far from a uniform distribution, with cinematic_ambient associated with the greatest number of moods:



With respect to sentiment, there is a clear imbalance within the dataset: of the 204 unique genre/subgenre/mood combinations, 104 are "positive," 51 are "neutral," and 49 are "negative." The ramifications of this skewed distribution will be addressed below.

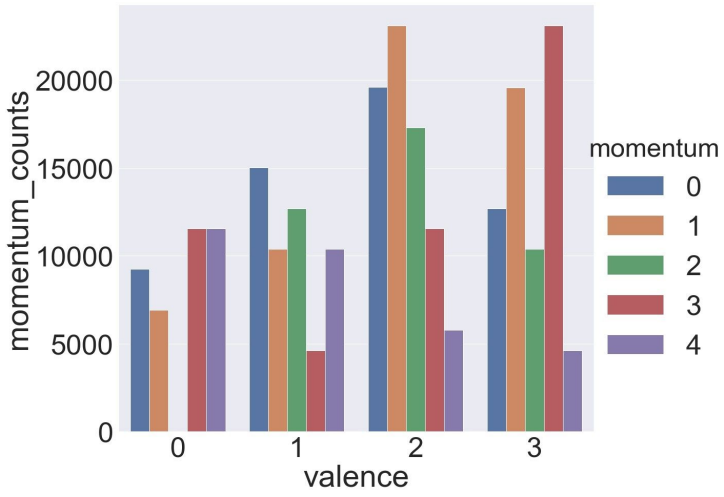
FEATURE EXTRACTION

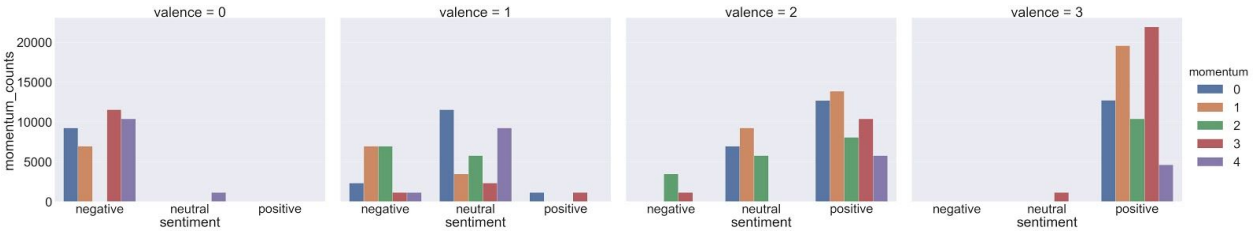
After experimenting with different quantities and combinations of acoustic features, the set was limited to 21 salient parameters, comprising three Mel frequency cepstral coefficients (MFCCs), three principal component analysis (PCA) coefficients, and 15 standard spectral and temporal features. Feature extraction was executed via the LibROSA Python audio/music analysis package. Each audio file in the dataset was analyzed in 10 second windows with a two second stride.[3] As such, the most recently-generated post-extraction input CSV consists of ca. 240,000 rows.

Given the highly subjective nature of mood selection/assignment, and the varying degrees of proximity of moods to each other, **valence** and **momentum,** hand-engineered ordinal categorical features associated with each mood were then assigned to each respective descriptor. Initially, the valence scale consisted of 4 levels, while the momentum scale consisted of 5 (each level corresponding to relative activity level and event-density).

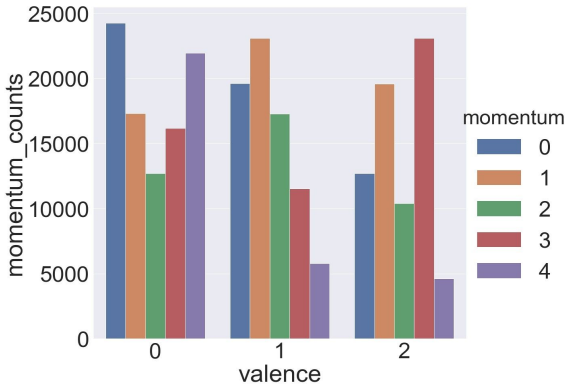
[3] These are fairly standard settings, enabling the classifiers to capture full phrases, while overlapping sufficiently to compensate for the truncation of any event in the previous window, and to detect structural changes accurately.

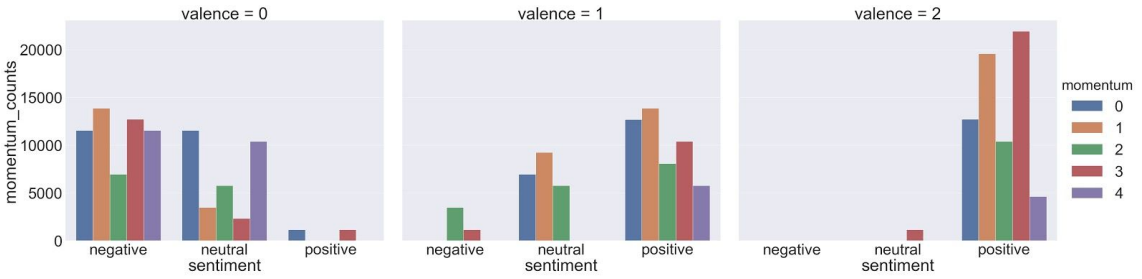
The bar graphs below illustrate relationships amongst sentiment, valence and momentum within the dataset:



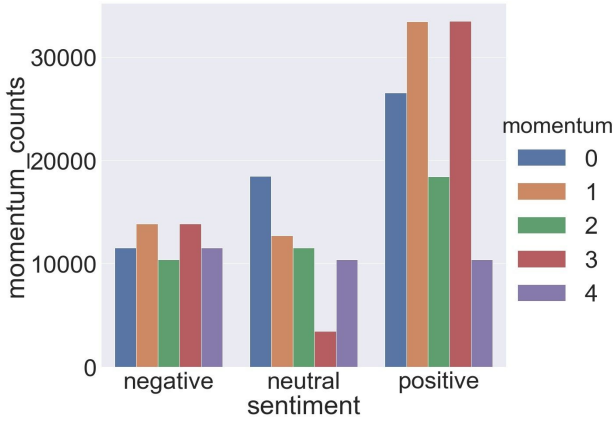


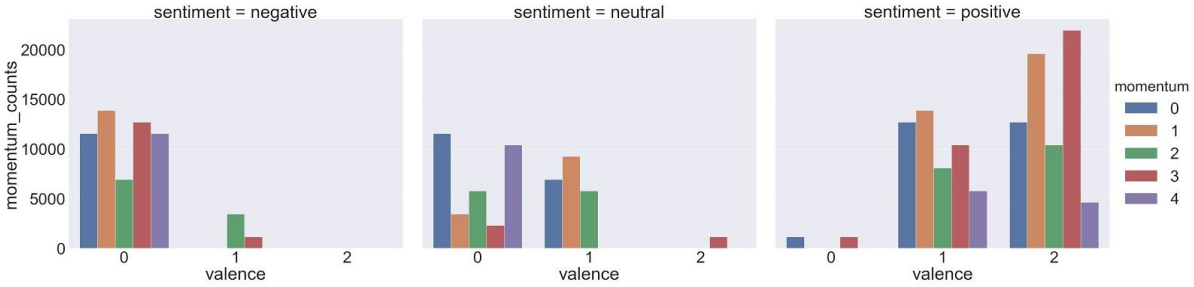
Given valence distribution imbalances, it was determined that valence levels 0 and 1 should be merged, thereby collapsing the valence scale into 3 steps. Here are the same relationships post-rescaling:





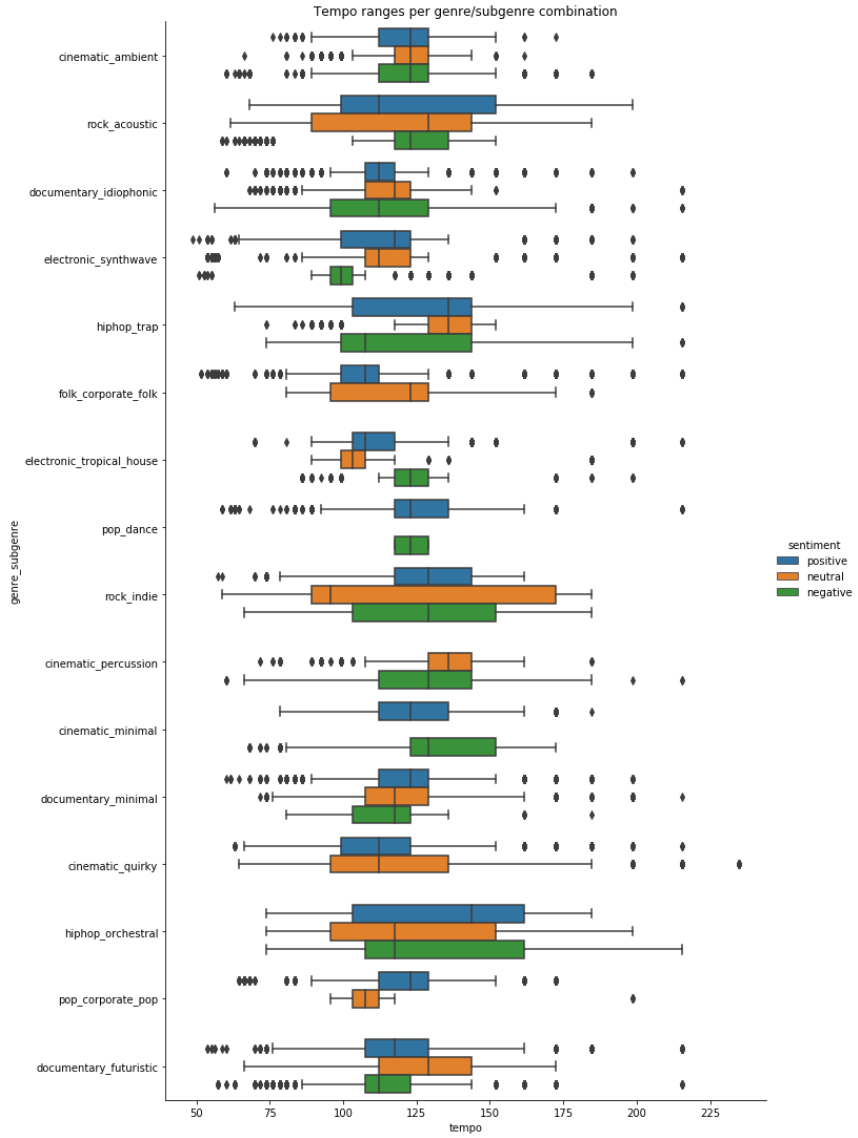
Although valence, like sentiment, has three levels, valence and sentiment demonstrate mutual independence (albeit moderately strong positive correlation). Likewise, momentum is independent from both sentiment and valence.





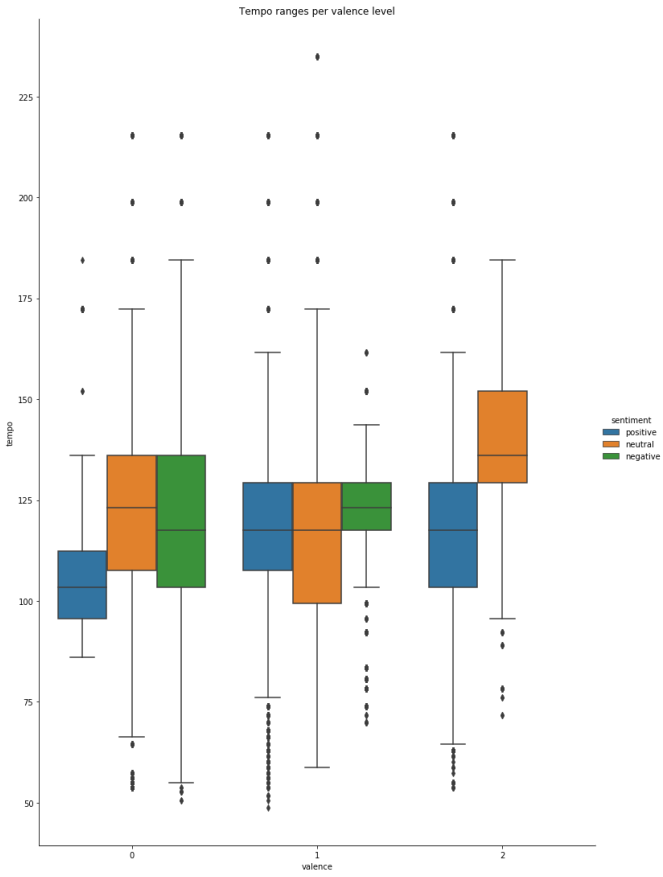
As tempo is amongst the extracted features, it is possible to examine the relationship between tempo and genre, subgenre, sentiment, valence, and momentum. As a baseline, the minimum (induced) tempo in the dataset is ca. 49 BPM, the maximum is 234, the mean is 120, and the median is ca. 118.

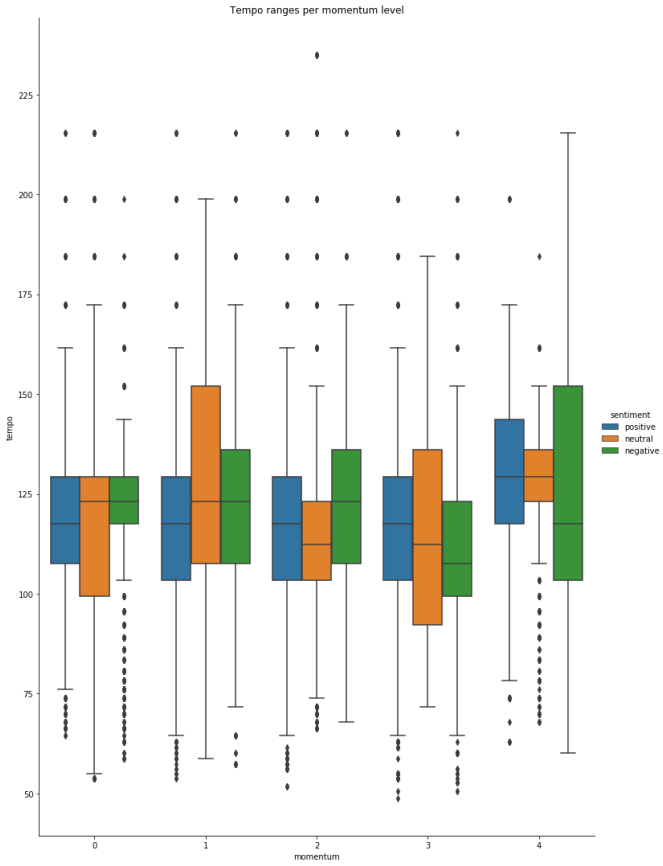
In the box plot below, tempo ranges for each genre/subgenre combination at each respective sentiment are depicted:



Most genre/subgenre combinations have median tempi at ca. 120 BPM. Tempo and sentiment are not clearly correlated. Hiphop_orchestral (positive sentiment) has the highest median tempo, and rock_indie (negative sentiment). There is no clear strong correlation between tempo and sentiment.

Here are the ranges for each valence and momentum level, again segmented into sentiment categories:





The highest valence and momentum levels have slightly higher median tempi than the other levels, but the correlation with sentiment is likewise unclear. There is no obvious linear increase in median tempo/tempo minimum and maximum across the remaining valence and momentum levels. In all three graphs, it is evident that there are generally more outliers at slower than faster tempi. (It is quite possible that the extreme upper range tempi represent miscalculations on the part of the LibROSA tempo-tracking function, as the maximum allowable tempo is 200 BPM). These observations indicate that sentiment, valence, and momentum are potentially meaningful features, in that distinctions amongst levels cannot simply be explained by tempo differences. Conversely, tempo proves to be a valid feature for the same reason.

MODELING

Descriptor, Sentiment, Valence, and Momentum Classification

As may be inferred from the above graphs, the data is hierarchical in nature. That is to say: each genre is connected to (the "parent" of) a subset of subgenres, and each subgenre to a subset of moods. By the same token, conditioning valence and momentum classification on sentiment would in principle improve model performance. As such, it was decided that: 1) binary (one-vs.-rest, or OVR) classifiers be applied to the top-level targets of genre and sentiment, and 2) lower-level target classifiers ("children") be conditioned

on their respective "parents" (i.e., subgenre given genre, mood given subgenre, valence given sentiment, momentum given sentiment).

KNN vs. MLP

After testing several models, it was determined that K-Nearest Neighbors (KNN) and Multi-Layer-Perceptron (MLP) models demonstrated superior performance. For the MLP, a maximum iteration of 800 was necessary for the model to converge. Otherwise, all default scikit-learn hyperparameter settings (number of layers, optimizer, learning rate, etc.) were maintained.

For both models, a train/test split of 70/30 was selected, and stratification was applied to the test set to ensure that all classes would be represented. The full classification reports are available upon request, but here are the average accuracy scores:

Target	KNN	MLP
Binary Genre	**0.999**	0.990
Binary Sentiment	**0.992**	0.924
Subgenre	0.530	**0.535**
Mood	**0.983**	0.972
Valence	**0.995**	0.973
Momentum	**0.995**	0.943

From the summary statistics, there are two obvious results: 1) the KNN model's performance is superior to that of the MLP; and 2) accuracy scores are quite high for all targets except subgenre (the reasons for which remain unknown).

EXTERNAL TEST TRACKS: ICELANDIC INDIE COMPILATION

To date, numerous non-Amper tracks have been tested. In the scope of this project, I have focused on the compilation *This is Icelandic Indie Music*, Vol. 1 (Record Records, 2013), as well as the track "Crystals" off of Of Monsters and Men's album *Beneath the Skin* (Republic Records, 2015). These selections were made for two reasons: 1) the tracks included *mostly* represent genres supported by Amper (with two notable exceptions: one soul and one reggae track); and 2) out of convenience (i.e., they were in my iTunes library). It should be noted that these tracks were not labeled with relevant metadata.

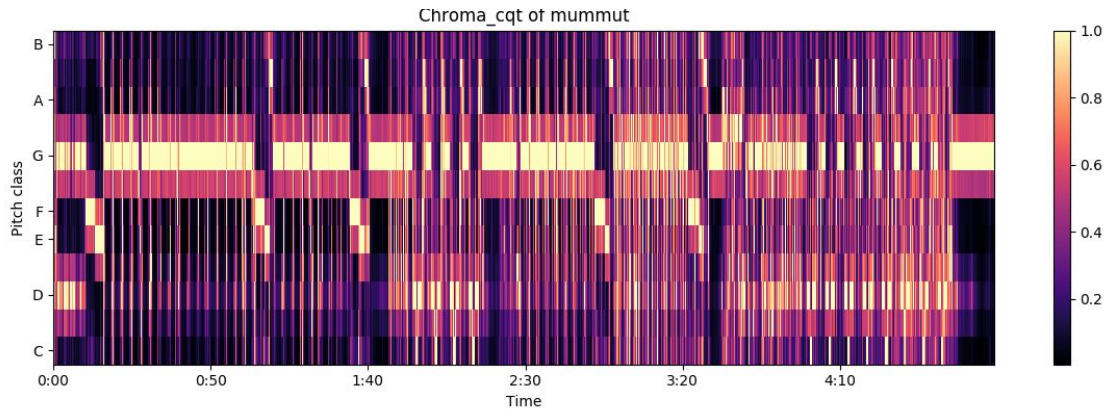
As was mentioned above, Amper renders do not contain melodic lines or vocals. In order to avoid introducing the human voice as a confounding factor, I stripped the tracks of vocals using the Deezer Spleeter source separation library.[4] No further modifications were made to these tracks.

TEMPO, TONIC, FORM, AND MODEL INTEGRATION

Tempo and Tonic

For tempo and tonic (root note) induction, LibROSA utilities were employed. In the case of tempo, onsets were detected and static tempi inferred for each 10-second window. A chroma CQT (constant Q-transform) algorithm was applied, such that the most prominent chromatic pitch for each 10-second window would be returned. In both cases, a stride of 2 seconds was selected (for the sake of consistency with the descriptor classification output).

Below is an example of a CQT plot for an input track with a quite stable tonic:



Form Segmentation

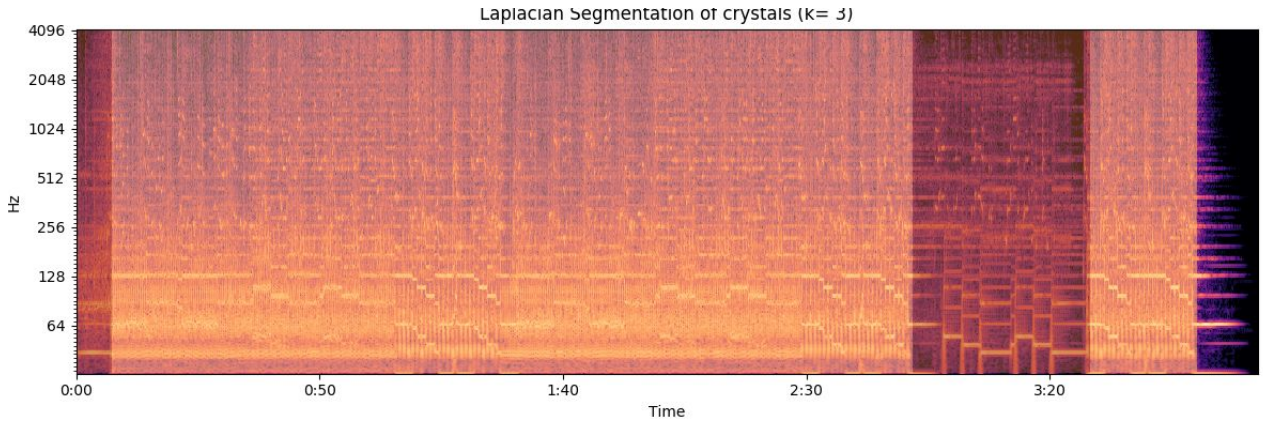
A Laplacian segmentation algorithm, which detects formal boundaries based upon analysis of the audio signal, was applied to perform a task that is otherwise subjective and ambiguous. (What cues signal the beginning of a new section? Change of texture, instrumentation, key, mood, onset density…:?)

For this procedure, in which onsets are clustered into segments via a KMeans clustering model, it is necessary to supply a value of *k.* But how does one select such a value? And when applying the model to multiple tracks, is it not possible that each will consist of a different number of discrete formal segments? In order to avoid arbitrary decision-making, I applied silhouette analysis--essentially a "grid search" procedure for KMeans--to select *k* values for each respective input soundfile. The range of possible *k* values was constrained by soundfile duration lower and upper limits. (For instance: one would not expect

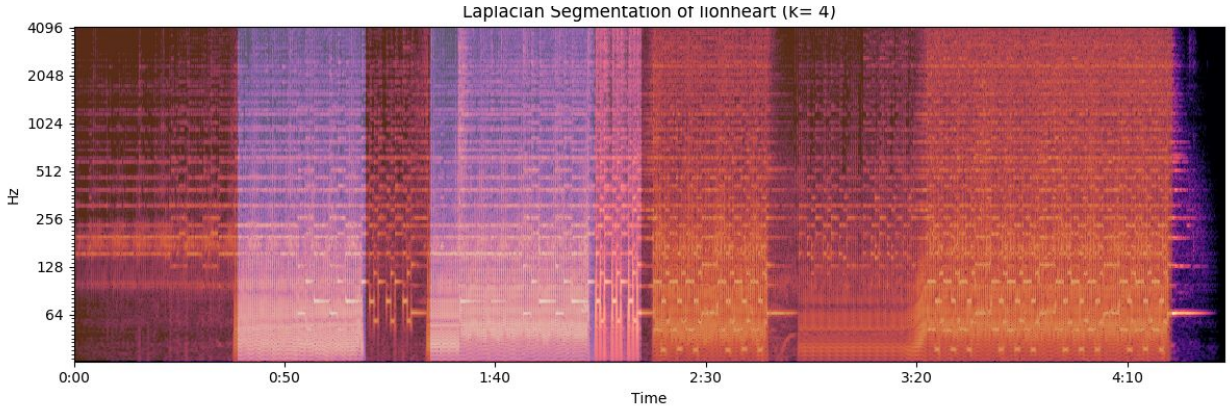
[4] I had been exploring existing accessible source separation options, and this utility has thus far proven to deliver the best performance.

a 10-second file to consist of 5 sections, and for a 4-minute file to consist of one or two sections would constitute a trivial application of form segmentation).

Below is an example of a form segmentation diagram, in which an introduction, main material, interlude (in which the introduction material or texture returns) and ending are inferred:



Here is a somewhat more complex example, with four form-segment types:



Merging and Reduction

Once all models had been applied to the input tracks, the model output labels for each 2-second increment were merged, as is depicted below:

COMPLETE_DATA_icelandic_indie

	Unnamed: 0	title	timestamp	genre	subgenre	mood	sentiment	valence	momentum	combined	tempo	key	form_segment	new
0	0	crystals	0	electronic	tropical_house	determined	negative	1	1	electronic_tropical_house_determined	136	F	1	new
1	1	crystals	2	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	136	F	1	
2	2	crystals	4	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	136	C	1	
3	3	crystals	6	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	1	
4	4	crystals	7	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	new
5	5	crystals	8	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	
6	6	crystals	10	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	
7	7	crystals	12	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	
8	8	crystals	14	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	
9	9	crystals	16	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	
10	10	crystals	18	electronic	synthwave	brooding	negative	0	3	electronic_synthwave_brooding	129	C	2	
11	11	crystals	20	electronic	synthwave	fun	negative	0	3	electronic_synthwave_fun	129	C	2	
12	12	crystals	22	electronic	synthwave	brooding	negative	0	3	electronic_synthwave_brooding	129	C	2	
13	13	crystals	24	electronic	synthwave	fun	positive	3	3	electronic_synthwave_fun	129	C	2	
14	14	crystals	26	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	136	C	2	
15	15	crystals	28	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	129	C	2	
16	16	crystals	30	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	129	F	2	
17	17	crystals	32	cinematic	percussion	driving	negative	0	4	cinematic_percussion_driving	129	A	2	
18	18	crystals	34	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	129	A	2	
19	19	crystals	36	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	136	G	2	
20	20	crystals	38	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	129	F	2	
21	21	crystals	40	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	136	F	2	
22	22	crystals	42	cinematic	percussion	driving	negative	0	4	cinematic_percussion_driving	136	F	2	
23	23	crystals	44	cinematic	percussion	driving	negative	0	4	cinematic_percussion_driving	136	F	2	
24	24	crystals	46	cinematic	percussion	driving	neutral	1	4	cinematic_percussion_driving	129	F	2	
25	25	crystals	48	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	129	G	2	
26	26	crystals	50	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	129	G	2	
27	27	crystals	52	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	129	G	2	
28	28	crystals	54	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	129	F	2	
29	29	crystals	56	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	129	F	2	
30	30	crystals	58	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	129	F	2	
31	31	crystals	60	hiphop	trap	celebratory	positive	3	4	hiphop_trap_celebratory	136	C	2	
32	32	crystals	62	pop	corporate_pop	exciting	positive	3	4	pop_corporate_pop_exciting	136	C	2	
33	33	crystals	64	pop	corporate_pop	exciting	positive	3	4	pop_corporate_pop_exciting	136	C	2	
34	34	crystals	66	pop	corporate_pop	exciting	positive	1	4	pop_corporate_pop_exciting	136	C	2	
35	35	crystals	68	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	129	C	2	
36	36	crystals	70	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	129	C	2	
37	37	crystals	72	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	136	C	2	
38	38	crystals	74	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	129	C	2	
39	39	crystals	76	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	136	C	2	
40	40	crystals	78	pop	corporate_pop	exciting	positive	1	3	pop_corporate_pop_exciting	136	C	2	

In the interest of 1) observing dominant classification trends for each inferred formal segment of each track and 2) reshaping the data in such a way that it would be conducive to creating a timeline from which renders could be generated (details to follow), I created a CSV consisting of the most frequently occurring targets (genre, subgenre, mood, tempo, and key) for each form segment, and calculated each segment's duration. Furthermore, through consulting the previously-mentioned active descriptor dictionary, I created a column for descriptor names corresponding to the identified genres, subgenres, and moods (a necessary step for timeline generation):

majority_rules_mlp_icelandic_indie

	title	duration	combined	sentiment	valence	momentum	tempo	key	gt_sentiment	descriptor
0	agent_fresco	177	electronic_synthwave_ominous	neutral	1	4	123	F#/Gb	negative	edm_synthwave_ominous
1	agent_fresco	109	documentary_idiophonic_relaxed	positive	1	3	123	F#/Gb	positive	documentary_idiophonic_relaxed
2	crystals	7	electronic_synthwave_exciting	negative	0	3	136	C	positive	edm_synthwave_exciting
3	crystals	165	electronic_synthwave_exciting	positive	1	3	129	C	positive	edm_synthwave_exciting
4	crystals	35	documentary_idiophonic_idle	positive	1	3	129	F	neutral	documentary_idiophonic_idle
5	crystals	23	electronic_synthwave_exciting	positive	1	3	136	C	positive	edm_synthwave_exciting
6	ensimi	22	documentary_idiophonic_happy	positive	1	1	112	D	positive	documentary_idiophonic_happy
7	ensimi	137	electronic_synthwave_confident	positive	1	3	144	A	positive	edm_synthwave_confident
8	ensimi	13	hiphop_trap_celebratory	positive	1	3	136	D	positive	hip_hop_trap_inspirational_celebratory
9	ensimi	56	electronic_synthwave_exciting	positive	1	3	136	A	positive	edm_synthwave_exciting
10	fm_belfast	7	documentary_idiophonic_sad	neutral	0	4	129	D	negative	documentary_idiophonic_sad_thick
11	fm_belfast	166	electronic_synthwave_sentimental	positive	1	3	129	G	neutral	edm_synthwave_sentimental
12	kirayama_family	16	cinematic_quirky_heartfelt	positive	1	1	129	B	neutral	cinematic_quirky_heartfelt
13	kirayama_family	180	documentary_futuristic_driving	positive	1	3	129	C#/Db	neutral	documentary_futuristic_driving
14	lionheart	39	documentary_minimal_documentary_wondrous	positive	1	1	129	G	positive	documentary_minimal_wondrous
15	lionheart	30	pop_dance_uplifting	positive	2	2	129	D#/Eb	positive	pop_dance_uplifting
16	lionheart	16	pop_dance_dreamy	positive	1	1	129	D#/Eb	positive	pop_dance_dreamy
17	lionheart	39	pop_dance_confident	positive	2	3	129	D#/Eb	positive	pop_dance_confident
18	lionheart	11	hiphop_trap_celebratory	positive	2	4	123	C	positive	hip_hop_trap_inspirational_celebratory
19	lionheart	3	electronic_synthwave_wondrous	positive	1	3	123	G#/Ab	positive	edm_synthwave_wondrous
20	lionheart	28	electronic_synthwave_wondrous	positive	1	3	123	G#/Ab	positive	edm_synthwave_wondrous
21	lionheart	37	cinematic_quirky_wondrous	positive	1	3	129	D#/Eb	positive	cinematic_quirky_wondrous
22	lionheart	58	electronic_synthwave_wondrous	positive	1	3	129	G#/Ab	positive	edm_synthwave_wondrous
23	lockerbie	43	documentary_futuristic_melancholic	negative	0	0	117	G	negative	documentary_futuristic_melancholic
24	lockerbie	65	cinematic_ambient_angelic	positive	2	1	89	D	positive	cinematic_ambient_angelic
25	moses_hightower	6	documentary_futuristic_determined	positive	1	2	99	G	negative	documentary_futuristic_determined
26	moses_hightower	134	electronic_tropical_house_relaxed	positive	1	3	99	D#/Eb	positive	edm_house_tropical_relaxed
27	moses_hightower	5	electronic_synthwave_ominous	positive	1	0	99	D#/Eb	negative	edm_synthwave_ominous
28	moses_hightower	11	electronic_synthwave_ominous	positive	1	3	99	G	negative	edm_synthwave_ominous
29	moses_hightower	21	documentary_idiophonic_dreamy	positive	1	3	99	E	positive	documentary_idiophonic_dreamy
30	moses_hightower	2	electronic_tropical_house_longing	positive	1	3	99	D#/Eb	negative	edm_house_tropical_longing
31	moses_hightower	34	electronic_synthwave_funky	positive	1	3	99	D#/Eb	positive	edm_synthwave_funky
32	moses_hightower	16	electronic_synthwave_sentimental	positive	1	3	99	C#/Db	neutral	edm_synthwave_sentimental
33	mummut	17	documentary_minimal_documentary_wistful	neutral	0	2	161	G	positive	documentary_idiophonic_serious_thin
34	mummut	96	hiphop_trap_menacing	positive	1	3	161	G	negative	hip_hop_trap_mysterious_menacing
35	mummut	25	electronic_synthwave_exciting	positive	1	3	161	G	positive	edm_synthwave_exciting
36	mummut	98	pop_corporate_pop_exciting	positive	1	3	161	G	positive	corporate_pop_exciting_high_energy
37	mummut	51	electronic_synthwave_exciting	positive	1	3	152	D	positive	edm_synthwave_exciting
38	ojba_rasta	3	hiphop_trap_celebratory	positive	1	3	136	C	positive	hip_hop_trap_inspirational_celebratory
39	ojba_rasta	172	electronic_synthwave_fun	positive	1	3	144	C	positive	edm_synthwave_fun
40	ojba_rasta	14	hiphop_trap_brooding	positive	1	1	136	C	negative	hip_hop_trap_tense_brooding
41	ojba_rasta	69	hiphop_trap_celebratory	positive	1	3	144	C	positive	hip_hop_trap_inspirational_celebratory
42	retro_stefson	8	electronic_tropical_house_carefree	neutral	0	4	123	G	positive	edm_house_tropical_carefree
43	retro_stefson	94	electronic_tropical_house_groovy	positive	1	3	123	E	positive	edm_house_tropical_groovy
44	retro_stefson	33	pop_dance_groovy	positive	1	3	123	E	positive	pop_dance_groovy
45	retro_stefson	61	electronic_synthwave_funky	positive	1	3	123	E	positive	edm_synthwave_funky
46	sykur	16	documentary_futuristic_driving	positive	2	3	123	D#/Eb	neutral	documentary_futuristic_driving
47	sykur	102	documentary_futuristic_frenetic	positive	1	3	123	C	negative	documentary_futuristic_frenetic
48	sykur	16	electronic_synthwave_confident	positive	1	3	123	D#/Eb	positive	edm_synthwave_confident
49	sykur	76	documentary_futuristic_frenetic	positive	1	3	123	C	negative	documentary_futuristic_frenetic
50	sykur	14	documentary_idiophonic_idle	positive	1	2	123	G	neutral	documentary_idiophonic_idle

Classifier Model Selection

Although the KNN model outperformed the MLP on Amper test data, the MLP tended to yield more accurate genre and subgenre classifications for the Icelandic indie data. This could be discerned once the above redux was generated for both classification models. In general, the style predictions for the MLP were more on-target. More specifically, when any of the classification models tested were unable to make an accurate prediction, and/or became too sensitive to a dominant drum track, they would default to "cinematic_percussion," a genre/subgenre combination lacking in any instruments generating precise pitches. For this dataset, the KNN was outputting "cinematic_percussion" often enough for it to be a top-ranked candidate for a number of formal segments. This was never the case for the MLP. With the exception of the occasional brief drum fill, the tracks themselves never exhibited any purely percussive passages.

RENDER GENERATION

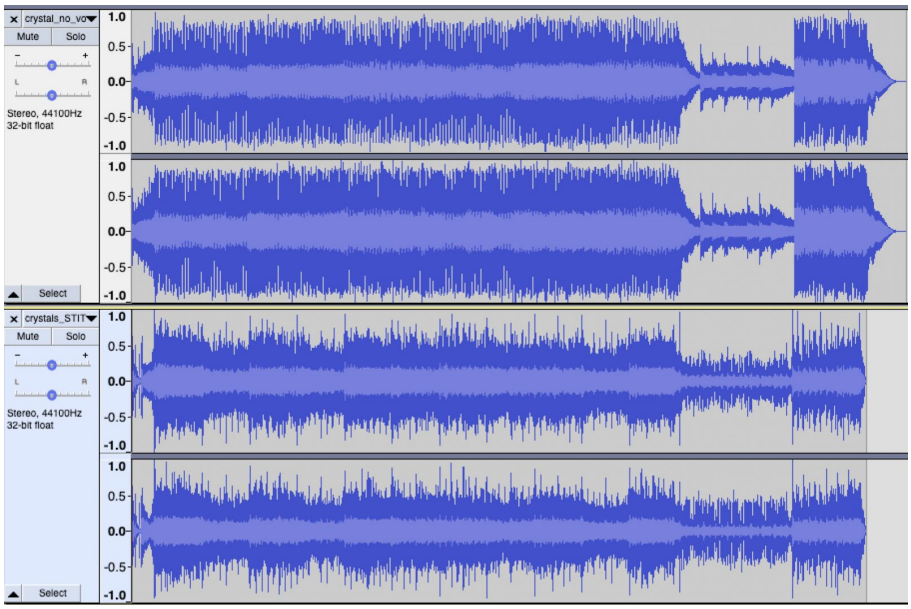
As was explained in the Introduction, two sets of renders were generated for each input track, using values contained in the "majority_rules_mlp_icelandic_indie" CSV above: 1) single-region renders, each representing one of the top-ranked descriptors, tempi, and tonics for the given track; 2) multi-region (or "stitched") renders, the number of regions being determined by the number of induced formal segments. In both cases, instrumentation was randomized, for the purpose of producing heterogeneous output. As no single render is an ideal representation of a given descriptor, each descriptor in the single-region case was rendered at each top-ranked tempo in each top-ranked key, while in the multi-region case, ten renders were generated for each input track (each with a different instrumentation).

In order to create a render, it is necessary to enter data in a timeline JSON format, as depicted below:


```
"timeline": {
                    "spans": [
                        {
                            "actions": [
                                {
                                    "time": 0,
                                    "add_region": {
                                        "key": {
                                            "tonic": "C"
                                        },
                                        "id": 222,
```

```
                                        "cut_at": 48,
                                                                    "descriptor":
"documentary_idiophonic_idle"
                                    }
                                }],
                            "time": 0,
                            "tempo": 136.0,
                            "id": 111,
                            "type": "metered",
                                "instrument_groups": [{"instrument_group":
"grand_piano"}]
                        },
                        {
                            "time": 344,
                            "type": "unmetered"
                        }
                    ]
                }
```

In the figure below, the original input track “Crystals” has been superimposed on a corresponding multi-region render. As one may observe, the region boundaries line up exactly (i.e., with significant shifts in amplitude reflecting changes in texture/activity levels):



To listen to the original (without vocals), click here and here.

SINGLE-REGION RENDER CLASSIFICATION AND MULTI-REGION RENDER COMPARISON

In the following table are the accuracy scores for the single-region renders:

Classifier	**Accuracy Score**
Genre	0.897
Subgenre	0.858
Mood	0.715
Sentiment	0.821
Valence	0.705
Momentum	0.752

For the multi-region renders, the most frequently occurring value for each target variable for each formal segment amongst each set of ten examples was compared to the corresponding value for the respective input track. This table represents the *mean* accuracy scores for multi-region renders across titles.

Classifier	**Mean Correlation Score**
Genre	0.900
Subgenre	0.855
Mood	0.755
Sentiment	0.635
Valence	0.686
Momentum	0.413

Genre accuracy scores by title:

[('agent_fresco', 1.0), ('crystals', 1.0), ('ensimi', 0.75), ('fm_belfast', 0.5), ('kirayama_family', 1.0), ('lionheart', 1.0), ('lockerbie', 1.0), ('mummut', 1.0), ('ojba_rasta', 0.75), ('retro_stefson', 1.0)]

Subgenre accuracy scores by title:

[('agent_fresco', 1.0), ('crystals', 1.0), ('ensimi', 0.75), ('fm_belfast', 0.5), ('kirayama_family', 1.0), ('lionheart', 1.0), ('lockerbie', 1.0), ('mummut', 0.8), ('ojba_rasta', 0.75), ('retro_stefson', 0.75)]

Mood accuracy scores by title:

[('agent_fresco', 1.0), ('crystals', 1.0), ('ensimi', 0.5), ('fm_belfast', 0.5), ('kirayama_family', 1.0), ('lionheart', 1.0), ('lockerbie', 1.0), ('mummut', 0.8), ('ojba_rasta', 0.25), ('retro_stefson', 0.5)]

Sentiment accuracy scores by title:

[('agent_fresco', 0.5), ('crystals', 0.5), ('ensimi', 1.0), ('fm_belfast', 0.0), ('kirayama_family', 0.0), ('lionheart', 1.0), ('lockerbie', 1.0), ('mummut', 0.6), ('ojba_rasta', 1.0), ('retro_stefson', 0.75)]

Valence accuracy scores by title:

[('agent_fresco', 0.0), ('crystals', 1.0), ('ensimi', 0.75), ('fm_belfast', 0.5), ('kirayama_family', 1.0), ('lionheart', 0.5555555555555556), ('lockerbie', 1.0), ('mummut', 0.8), ('ojba_rasta', 0.25), ('retro_stefson', 0.5)]

Momentum accuracy scores by title:

[('agent_fresco', 0.0), ('crystals', 0.5), ('ensimi', 0.5), ('fm_belfast', 0.0), ('kirayama_family', 0.0), ('lionheart', 0.7777777777777778), ('lockerbie', 0.5), ('mummut', 0.6), ('ojba_rasta', 0.75), ('retro_stefson', 0.5)]

For both the single-region and multi-region renders, scores for genre and subgenre are reasonably high and comparable. Mood scores are relatively low in both cases. While sentiment classification remains in a reasonable range in the former, it is surprisingly low in the latter. Most alarming is the low momentum score for multi-region renders.

As may be observed from the multi-region score distributions, there is a high degree of variance amongst titles across all target variables except genre and subgenre. This variance, as well as the relatively low mean scores, can be explained by several factors, but one crucial factor is the sentiment class imbalance mentioned in the EDA section. Sentiment misclassifications would in turn influence valence and momentum classification accuracy, as the latter are conditioned on the former. Another consideration is the multi-region render sample size. It would be worth repeating the comparison procedure with a greater number of multi-region renders per input track.

CONCLUSION AND FUTURE WORK

This project constituted the initial phase in an endeavor to generate Amper renders based upon feature extraction and classification of music data "from the wild." Through fusing descriptor classification, tempo and tonic induction, and form segmentation models, it was possible to generate single-region and multi-region renders that embodied structural and stylistic characteristics of the input tracks. When presented with the renders, the classification models produced generally accurate predictions, with a few notable exceptions. The reasons for prediction anomalies are currently under investigation.

In the future, it would first and foremost be of interest to train the classifiers on more Amper data. (The resource limitations of render time and competing server requests impose perennial constraints on dataset size.) Along similar lines, it is intended to expand from the sandbox of the Icelandic indie compilation to the Free Music Archive (FMA) dataset. This would be advantageous both given the dataset size, and the accompanying metadata.[5] As distances amongst genres, subgenres, and moods are not equal, establishing a descriptor space distance metric would be instructive in establishing a classifier weighting scheme. These weights could be learned and would inform optimization. In addition, such a scheme would allow for an "UNK" class for genre, subgenre, and mood target variables. Eventually, descriptor classification and instrument recognition models will be integrated, and will likely fall into a symbiotic

[5] That said, there is still room for ambiguity: an FMA genre label may not be compatible with an Amper label for the same genre, and a track in the FMA dataset whose genre label is not reflected in Amper's database may be quite similar in nature to an existing Amper descriptor.

relationship--especially given the crucial role that instrumentation plays in differentiating descriptors from each other. Prior to these developments, however, a web-based classification/generation prototype app will be launched and tested within the company, as a means of validation and gathering feedback on the current state of the project.

ACKNOWLEDGEMENTS

The author wishes to acknowledge Cole Ingraham and Nate Moon for their efforts, insights, and support, and Adam Gardner for his pioneering contributions.