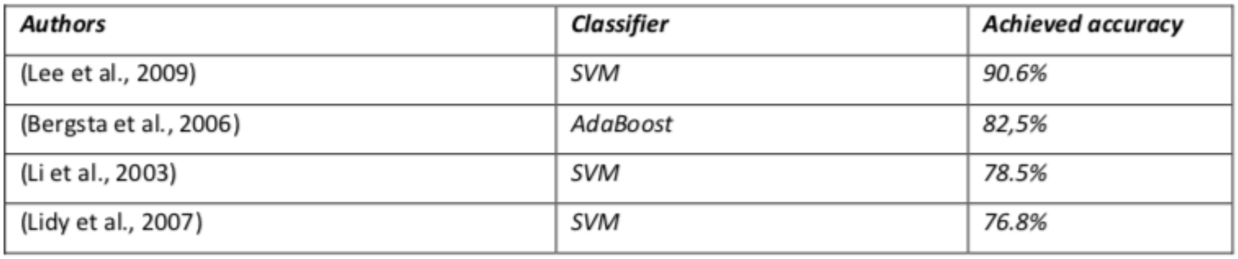
Automated Music Genre Classification

Benjamin Karahadian

Introduction

The purpose of this project is to expand established genre classification techniques to cover complex mixed-genre music. Throughout the literature, the datasets used are typically hand-picked to neatly fall into the relevant categories. For example, the GTZAN dataset, which is commonly used for both training and testing in genre classification projects, contains song clips that cleanly fall into ten relatively low-level macro-genres (blues, country, disco, rock, jazz, metal, etc.). While useful for proof-of-concept, this is unrealistic and will not scale for use in real-world scenarios. For example, the majority of Björk's music is a mixture of Pop and Electronica: classifying it solely in either category would be erroneous. Therefore, a sufficiently advanced algorithm needs the ability to predict multiple genres.

This project was inspired by, but does not directly expand upon, Valeri Tsatsishvili's "Automatic Subgenre Classification of Heavy Metal Music" (2011), wherein the author trains and tests on a dataset of pure-genre metal tracks, distinctive songs that fall cleanly into the relevant genres. The dataset contains 30 tracks for each of the metal genres considered: Black, Death, Gothic, Melodic Death, Power, and Progressive. The author uses the MIR toolbox in Matlab to extract an assortment of spectral features (i.e. the predictor variables) from the middle minute of each track. Classification is conducted using three methods: ADA-Boost/decision tree, K-Nearest Neighbors, and a custom hierarchical maximal margin classifier. The success rates for these methods never exceed 50%, so, instead of expanding on these models, I have focused on the author's successful references which utilized Support Vector Machines (SVM's) for 75+% success-rates on the aforementioned GTZAN dataset (see Figure 1). Therefore, I will be primarily employing SVM's in this project.



Authors	*Classifier*	*Achieved accuracy*
(Lee et al., 2009)	*SVM*	*90.6%*
(Bergsta et al., 2006)	*AdaBoost*	*82,5%*
(Li et al., 2003)	*SVM*	*78.5%*
(Lidy et al., 2007)	*SVM*	*76.8%*

Figure 1: Referenced Models and Their Accuracies

The practical utility of genre classification should be obvious for anyone who frequents online music shops or streaming services. On these websites, music is typically browsable via tags, so that users can, through filters, tailor the presented set of music. Often, these tags are added by the artists or record labels (on relatively open platforms like Bandcamp) or by the site administrators (likely the case for most streaming services). In the former case, the tagger faces incentives to add extraneous tags so the music will appear on more filters; in the latter case, man-hours must be spent to classify music correctly by hand. Both of these problems could be fixed by an automated classification algorithm.

The Data

The dataset used in this project comes from my own personal music library, which is composed of .m4a files encoded at 320 kbps VBR. I have classified each song by hand and have made the following exclusions for ease of use.

First, I have purged the dataset of the "Folk Music" tag because it is not a real genre with any unifying musical traits. It is, instead, an umbrella term that contains ethnic music of any geographical region, whether it be Celtic, Slavic, Middle-Eastern, Asian, etc.. Even these more specific Folk Music terms may need to be further fractionated since each region will likely contain multiple genres, such as dances, dirges, and religious hymns. Needless to say, my knowledge of Folk Music becomes more and more insufficient as the genres become necessarily more specific. Additionally, if I did have the requisite knowledge, I would not have a large enough sample size of the specific genres to extract any useful conclusions from working with them. For these reasons, I have removed obvious Folk tracks (of which there are not many) and have reduced all Folk-imbued music to its constituent genres. For example, the music of Ensiferum is usually classified as Folk Metal, but, here, it has been classified as Melodic Death Metal and Power Metal, the fundamental genres remaining if stripped of folk melodies and themes.

Next, I have limited my definition of Classical Music, which arguably suffers the same necessity for fractionation, to strictly symphonic music. This encompasses stereotypical Classical Music such as ballets, symphonies, and film scores. While this division is possibly still overly-broad, it is necessary to exclude other acoustic/Classical music (which tend to be intros, interludes, and outros) because 1) I am unable to classify them sufficiently by hand, and 2) its overlap with Folk Music is often too complex to parse. This umbrella term was retained while Folk Music was excluded because its constituent sub-genres still contain musical similarities.

Another acceptable macro-genre I have used is Electronic. Electronic Music encompasses a vast array of constituent sub-genres, including EDM, IDM, Jungle, Drum 'n' Bass, House, and Techno. While stereotypical examples of these genres are easy to identify, most complex Electronic Music combines influences from many sub-genres. For example, consider the music of Aphex Twin: according to Wikipedia, *I Care Because You Do* contains elements of IDM, Ambient, Techno, and Trip Hop; *Richard D. James Album* contains elements of Electronica, IDM, Jungle, and Drill 'n' Bass; and *Syro* contains elements of IDM, Electro-Funk, Synth-Funk, and Acid Techno. To do this kind of parsing song-by-song, I would need to sink far more time into studying Electronic sub-genres than would be reasonable. Therefore, I have retained the Electronic umbrella term and have only distinguished the sub-genre of Ambient, primarily because it is very easy to identify and rarely mixes with other Electronic sub-genres within individual songs.

Further edits are the following. Alternative Rock has not been distinguished from Rock because there is little difference in overarching musical traits (the distinction is usually attributed to the scene of origin). Additionally, boundary-pushing "avant-garde" music (like Virus and Mr. Bungle) has been excluded because there are no mutual traits that join this experimental category into a cohesive genre. Also, genres that are only represented (in pure, un-mixed form) by less than three artists have been removed for lack of a significant sample size; these are: Blues, Darkwave, Deathcore, Djent, Dungeon Synth, Funk, Gothic Metal, Gothic Rock, Hip Hop, Industrial Rock, Metalcore, Neofolk, Post-Rock, and Psychedelic.

After these edits, the resulting dataset includes 5893 songs. Figure 2 depicts the distribution of songs across the genres for which this project's models will be testing. Note that mixed-genre songs are counted once for each of their constituent genres.

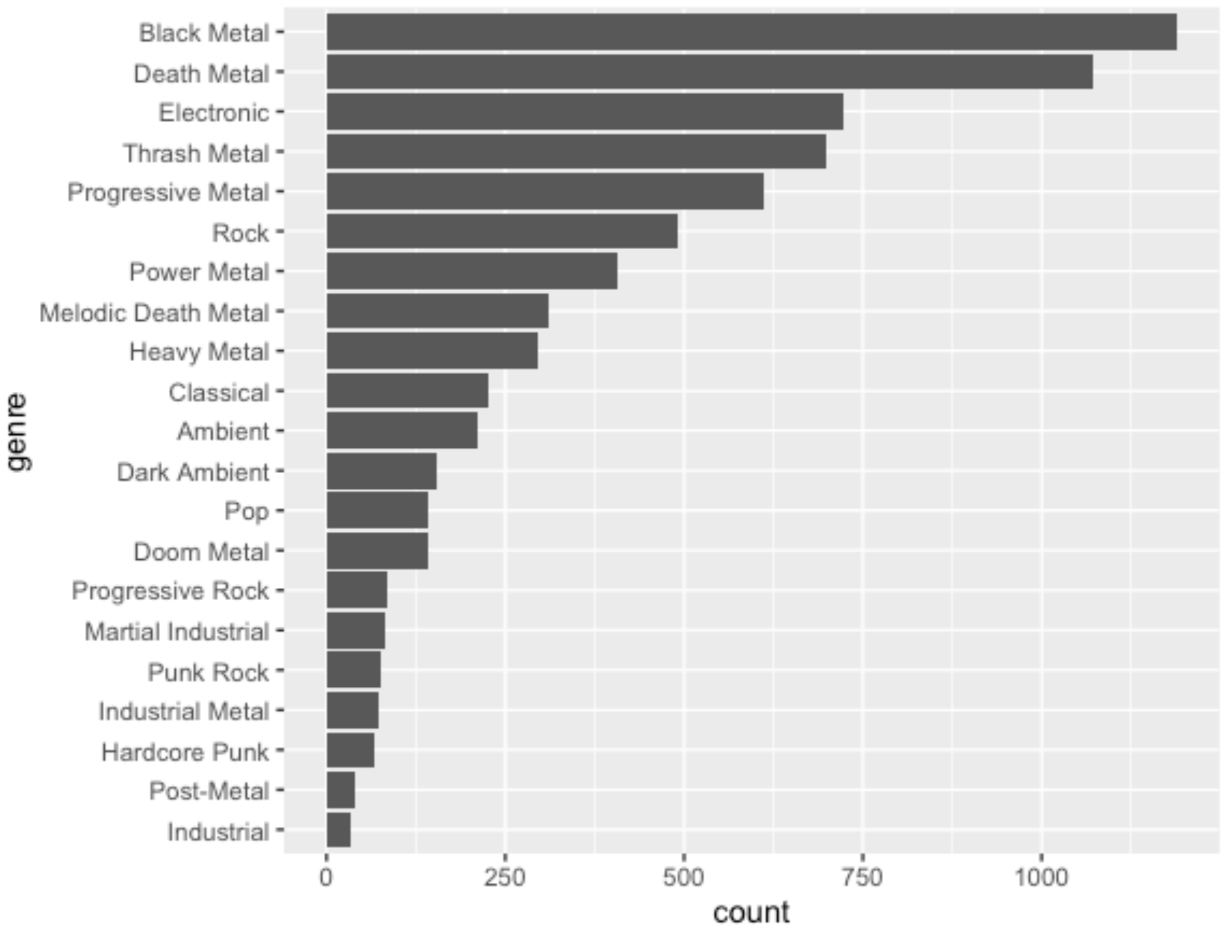


Figure 2: Genres and Their Sample Sizes

To generate the feature set for these data, I have used the Python package "Librosa". As per the default settings, the features of every 512th frame have been extracted for each song. This results in approximately 43 frames per second. Additionally, all silent frames have been removed; these could distort the predictions of the models since there is no information present. The final dataset, once loaded into R, is approximately 40 Gb.

The features are as follows: (Note that I am not an expert in signals so I cannot defend the merits of these features; they will be evaluated using a backwards selection process when modeling.)

- Spectral Centroid: measure of the location of the spectrum's center of mass
- Chromagram (1-12): measure of the energy present in each of twelve pitch categories
- MFCC (1-20): Mel-frequency cepstrum
- Polynomial Features (n = 1, 2, 3): the coefficients of the nth-order polynomial fitted to the columns of the spectrogram
- RMS: the root-mean-squared energy of each frame
- Spectral Rolloff (10% and 85%): the center frequency of a spectrogram bin under which a specific percent of the signal energy is located
- Spectral Bandwidth: a complex measure of the difference between the upper and lower frequencies
- Spectral Contrast: the mean energy of the peak frequency relative to the mean energy of the valley frequency for each spectrogram sub-band
- Spectral Flatness: a measure of noise-likeness (as opposed to tone-likeness)
- Tonnetz: the dimensions of each frame visualized on the tonnetz diagram
- Zero-Crossing Rate: the rate at which a signal crosses the axis

A tempo-/beat-tracking feature was not included because early exploration demonstrated that it was unreliable when extracting from music that lacks an obvious beat, Ambient and Dark Ambient music in particular. For example, raison d'être's "Katharos" registered at 152 bpm when there is hardly a beat-based tempo to be found.

Conceptual Considerations

There is a complex ontological relationship between the abstract concept of genre and the music that instantiates it. In most classification problems it is assumed that the considered categories have an absolute certainty to them, that a given prediction is either certainly correct or certainly incorrect. For example, if one were designing a spam filter for emails, any given prediction is either correct or incorrect because the definition of "spam" is built into the email as a defining set of features (sent by someone not in the contact list, regarding a subject of no relevance to the recipient, etc.) many of which the designer must create proxy variables to emulate. Musical genres are often implicitly regarded in this manner: that genre exists independently of any specific instance of music and that music correctly classified as such substantiates the abstract genre.

I do not believe this is how the phenomenon of genre operates. If one were to plot all extant music on a p-dimensional space, with p being the number of conceivable technical musical descriptors, the points would congregate around distinct areas. When these clusters reach a certain terminal density, a genre is formed. I believe this process is the *creation* of a genre, not the *substantiation* of an already-defined abstract concept.

For instance, the following graph plots Black Metal and Doom Metal bands on a much-simplified two-dimensional space with a speed axis (say, tempo or average notes per second) and a production quality axis (how noise-like is the sound). (Using more homogenous musical segments would be preferable as data points, but this simplification works for demonstration.) The Black Metal bands tend to cluster in the upper-left corner, while the Doom Metal bands tend to cluster in the lower-right corner. The circles symbolize the centroid whose dimensions would describe each genre's paradigmatic sound. Instead of the divisions in the genre space existing fundamentally and the music simply slotting in to their given positions within, I believe that the genres are created by the presence of the music itself.

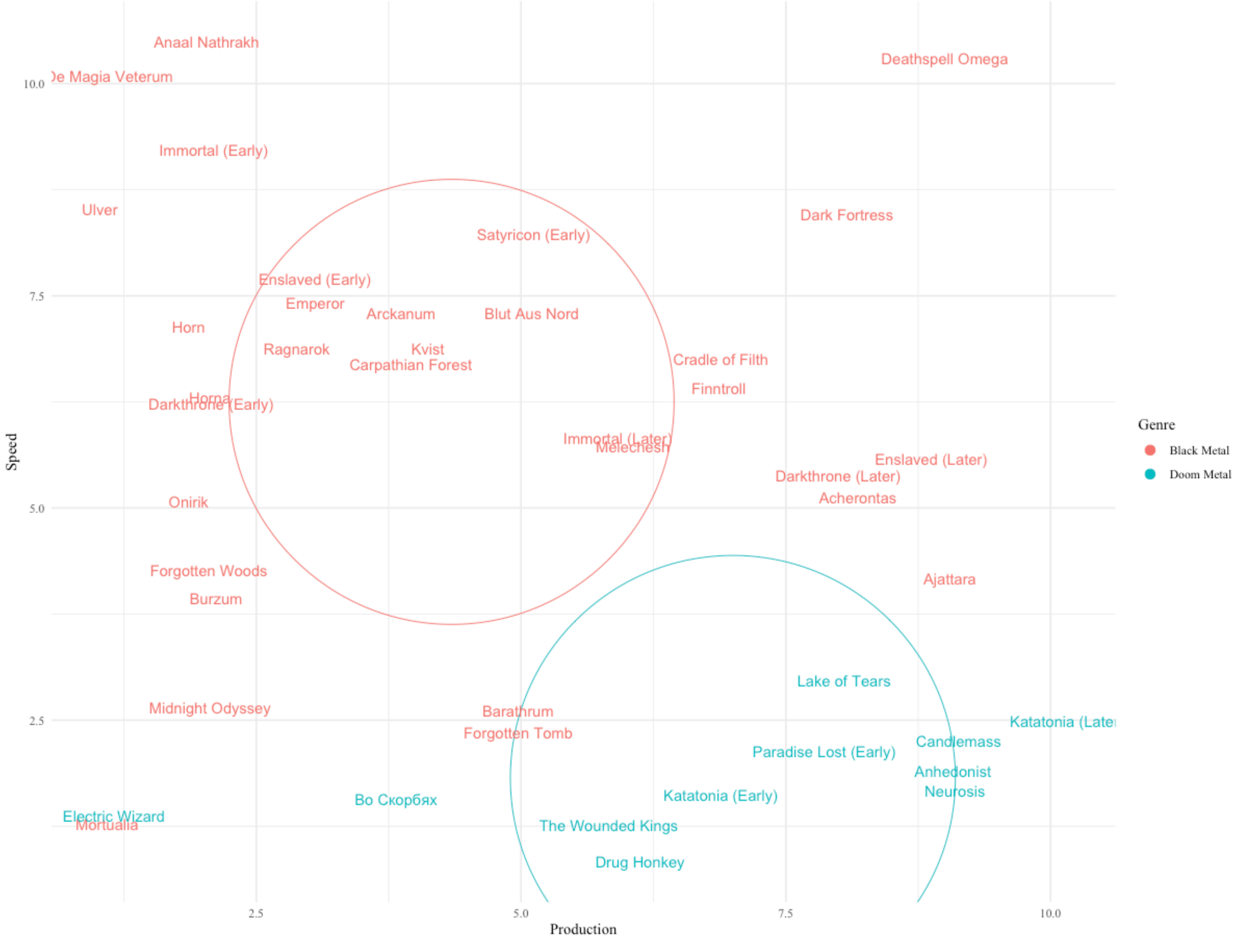


Figure 3: Simplified Genre-Space with Observations From Two Genres

What this entails is that the creation of new music has the potential to change the stereotypical sound of a genre or even pull already existing music into a new genre. For example, the music of

Meshuggah was originally classified as Technical/Progressive Thrash Metal; but once a host of similar bands emerged (such as Periphery and Born of Osiris), the term Djent was coined and retroactively reclassified Meshuggah's music as such.

What this means for automation is that even if an advanced system could be designed that could 1) compare all p-characteristics of a test point with all p-characteristics of all extant music, and 2) after each addition, reevaluate the feature space for areas that have newly reached the terminal density, a human hand would still be necessary to name any new genres and reclassify any training data that have changed designations before the algorithm can make a prediction.

Because the genre-space is always in flux, a human mind will always be necessary to update the algorithm. Therefore, pure automation of genre classification is impossible.

SVM Method Overview

Using the "e1071" package in R, three SVM variations were tested: linear, polynomial, and radial. For each Support Vector Machine there are two stages: first, the model is trained and tested on pure-genre music; and second, the model is adjusted and parameters are tuned to allow for multiple predictions, and then is tested on mixed-genre music.

The full dataset is not used for SVM modeling because my computational specifications are not up to the task. To create a reduced yet representative dataset, I have taken a random sample of twenty seconds worth of frames from each third of each track. In the case that a song is less than one minute in duration, the observation is not reduced. The resulting dataset is approximately 8 Gb: a significant decrease from the full set.

As noted previously, I have no prior knowledge that would allow me to remove redundant or irrelevant variables as they are extracted, therefore, first, a backwards elimination scheme was used to reduce the number of predictors. This process begins by setting aside a portion of the training data and setting a baseline for accuracy by generating a model using all predictors. Then, $p - 1$ models are fit using all combinations of $p - 1$ variables. Provided an increase in accuracy is achieved, the most successful of these smaller models is taken forward to the next stage. Here, $p_1 - 1$ models are fit where p_1 is the number of variables in the smaller model. This process is continued until further reduction in dimensionality yields no improvement. In this project, I have cross-validated the success-rates for each each model using two folds and have required three iterations of no improvement before the process terminates so as to guarantee that the global maxima in accuracy has been reached. Additionally, it should be noted that, with unlimited time and computing power, the parameters of the models should be tuned at each iteration of the elimination process; however, as I am unable to do this, I have worked under the assumption that the effect of untuned parameters is constant across each of the procedure's iterations.

Next, parameters for the SVM's are chosen using a tuning function (also found in the "e1071" package). These parameters are: cost for the linear model; cost, degree, and gamma for the polynomial model; and cost and gamma for the radial model. This tuning process uses cross-validation within another set-aside portion of the training data to pick the ideal parameters from a provided set. Once properly tuned, the model is evaluated on a test set of pure-genre songs and then on a complete set of pure- and mixed-genre songs.

Evaluating the model's success in classifying pure-genre songs is as simple as calculating the error-rate. To do so, all predictions for the individual frames of a particular song are collapsed into a frequency table and the most frequent prediction is selected. If this prediction matches the track's genre tag, then the model was successful; if it doesn't, then the model was unsuccessful. But once multiple genres are considered, the process becomes more complicated. This project considers up to three genres for each song, so two additional parameters must be validated: frequency values for when the top-two most frequent predictions are chosen and when the top-three are chosen. With multi-genre predictions, there are multiple possible semi-correct answers, and there must be deductions for misclassifications as well as missing classifications. The scoring method for the final evaluation is the following:

$$1 - n_{omit}(1/g) - n_{add}(1/p)$$

- n_{omit} : number of genres present in the true classification but not in the prediction
- n_{add} : number of genres present in the prediction but not in the true classification
- g : the number of true genres
- p : the number of predicted genres

This scoring mechanism ranges from -1 to 1, providing an intuitive evaluation: positive scores are mostly correct and negative scores are mostly incorrect.

Linear SVM

After conducting the feature selection process, the optimal linear model contains $62 - 3 = 59$ variables (see Figure 4). Note, however, that the increase in prediction power is relatively small: the optimal model yields approximately 1.5% better accuracy than the full model. The maxima at 3 variables removed is not spurious, though, as the same conclusion was generated with multiple repetitions and different fold compositions.

After being tuned and trained on pure-genre songs, the linear SVM's resulting success rate (on pure-genre songs) is **51.6%**. The breakdown for each of the considered genres is shown in Figure 5.1.

Classical and Electronic outperform most other genres by a significant margin, but this should not be surprising since these are the lowest-resolution genres available (as outlined previously) and are therefore relatively dissimilar to the rest. Although it is odd that Ambient, another highly distinct genre (in that an average listener could easily pick it out) performs relatively average. This is likely because the Ambient genre can be diverse in its instrumentation: from classic Kosmische Musik to more purely electronic ambient to more acoustic ambient.

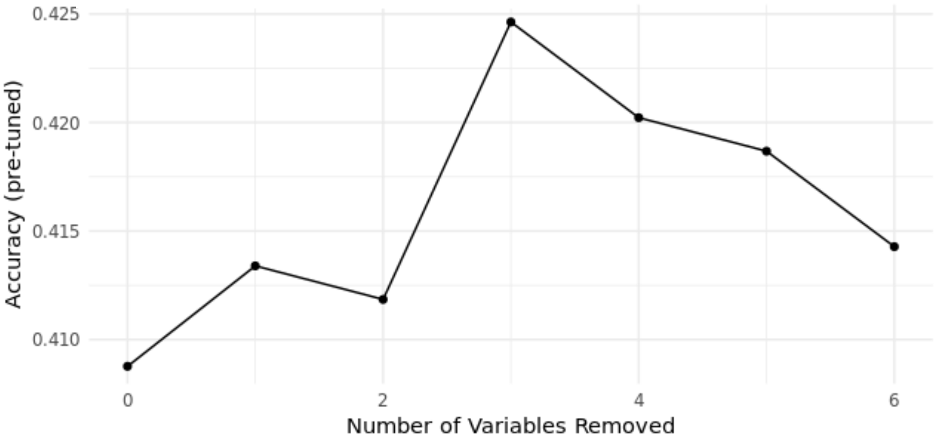


Figure 4: Linear SVM - Backward Elimination

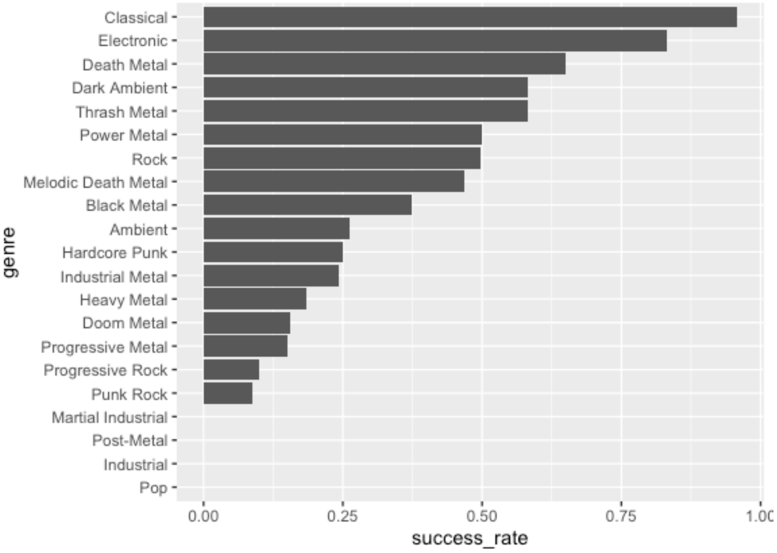


Figure 5.1: Linear SVM - Preliminary Success Rates

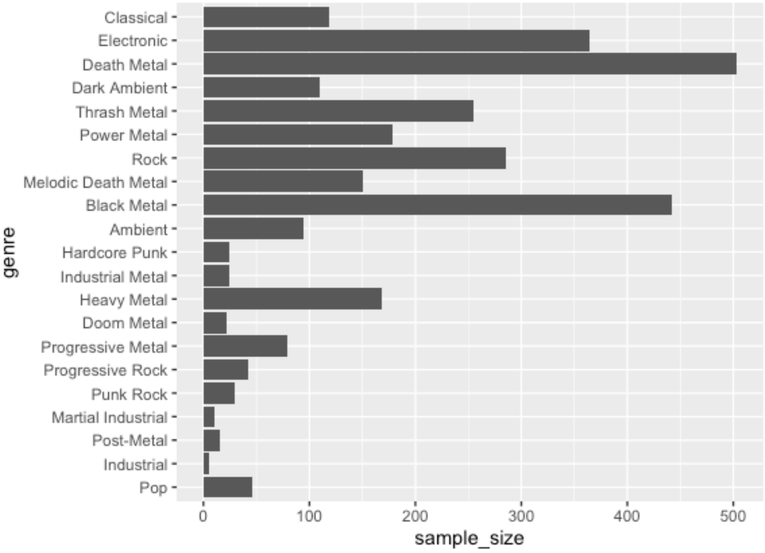


Figure 5.2: Linear SVM - Training Set Sample Sizes

What is surprising is that both Death Metal and Melodic Death Metal performed relatively well; if the latter was a derivative form of the former, it would be reasonable to expect the model to struggle in differentiating the two. This, however, is evidence that they are distinct genres, although there is clear evidence that they are related: they are each the most common erroneous prediction of the other (31% of erroneous predictions of Death Metal were Melodic Death Metal, and 36% of the reverse).

Figure 5.2 depicts the same genres and their respective sample sizes within the training set. Although not perfect, the correlation between sample size and greater predictive success is certainly significant. Outliers include Classical and Dark Ambient, which are unique enough sonically to perform well on smaller sample sizes, and Heavy Metal, which, being the original Metal genre, is easily conflated with primal versions of successive sub-genres (particularly Power Metal).

The model is then modified to predict multiple genres and is tested on a combined set of mixed-genre and pure-genre songs. Of note is that, after setting aside multiple portions of the original dataset to train the SVM and tune the different parameters, the resulting test set contains only 658 songs across many genres and diverse mixtures of genres; therefore, the resulting figures are rough estimates of the model's success on the global population of tracks. The average score is **-0.0481**. This means that, on average, the model classified songs partially correctly.

Figure 6 depicts the density of resulting scores for the mixed- and pure-genre songs within the test set. As expected, the pure songs were generally better classified, with less complete failures (-1), more complete successes (+1), and the maxima is located further up the score scale. This is likely because pure-genre songs are often only docked for additional predictions and not for a missing prediction: if the model predicts only one genre, then the vast majority of the constituent frames were classified as that genre, and one can be quite certain that the prediction is correct; if the model predicts multiple genres, there will necessarily be points docked for additional predictions. There are still a number of complete failures (-1), so there are still some pure-genre tracks with which the model was wholly confused.

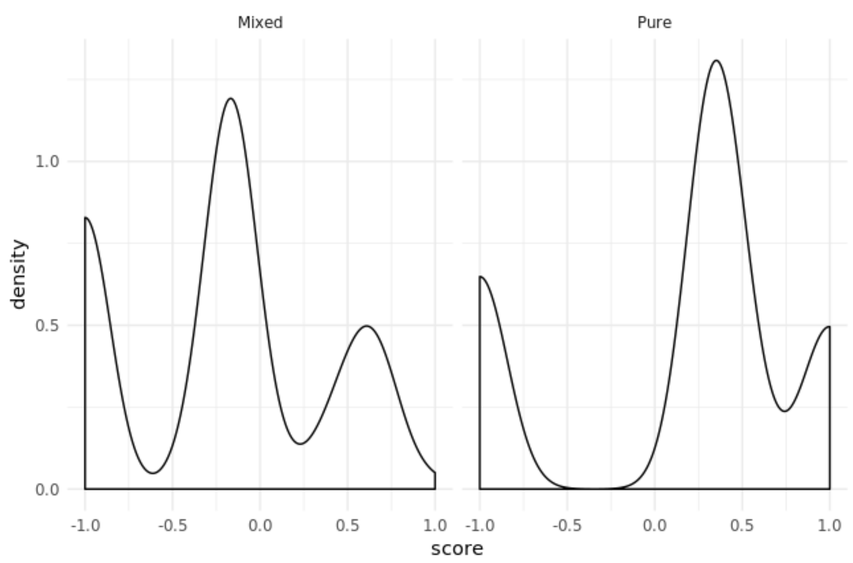


Figure 6: Linear SVM - Estimated Density Functions of Score

For the mixed-genre songs, it is promising that there is a noticeably dense portion of scores over 0.5. All of these predictions between 0.5 and 1 were scored at 0.667 and were songs with two true classifications and were predicted with three, in which only one was incorrect. In other words, even if the model's predictions were not perfect, they were still very successful with these tracks. This is evidence that accurately predicting mixed-genre music is genuinely possible, although my sample size somewhat tempers that statement.

Figure 7 provides each genre's proportion of correct detections in the full test set (combined mixed and pure) and in the mixed-genre portion only. Successful detection means that the model classified the song correctly in regards to that specific genre; for example, approximately 85% of mixed-genre songs that include Electronic influences were predicted to be at least partially Electronic.

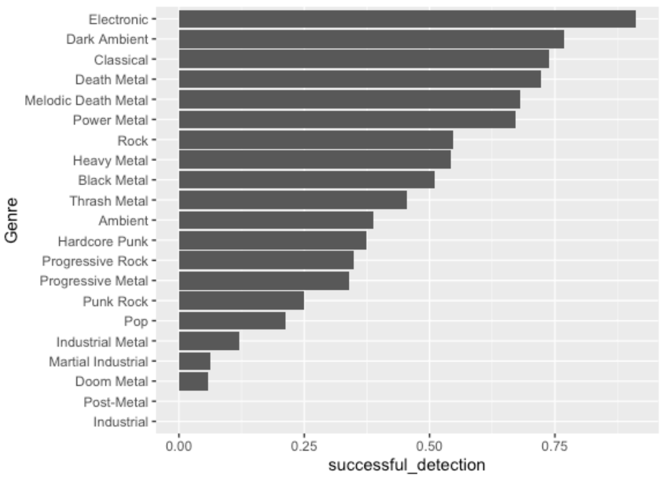


Figure 7.1: Linear SVM - Full Test Set

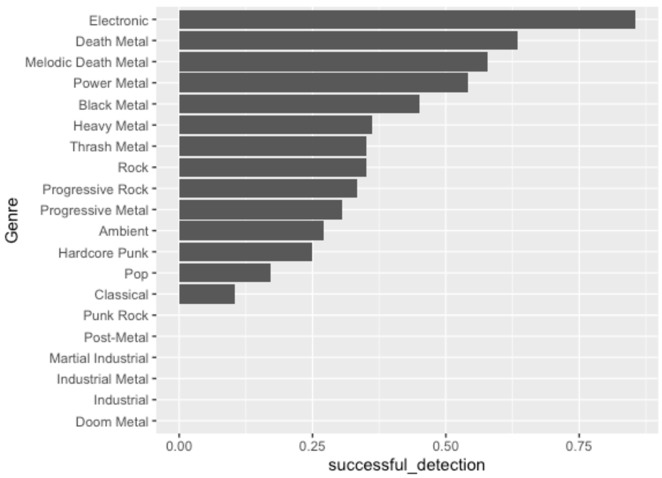


Figure 7.2: Linear SVM - Mixed-Genre Songs Only

Of note is that, upon consideration of multiple predictions, the success of Pop became non-zero. This is evidence that modern Pop music can justifiably be contained under the Electronic umbrella term, because all correctly-detected Pop tracks were classified as Electronic primarily and Pop secondarily. However, these two genres are are on two different levels of resolution, so it's not necessarily the case that these Pop tracks would be confused with other Electronic sub-genres like Jungle or IDM.

Additionally, the decline in the successful detection of Classical music is significant between the full test set and the mixed-genre portion. In fact, pure-genre Classical music in the test set was correctly predicted 100% of the time (although some had additional misclassification as Ambient or Dark Ambient). This indicates that, when Classical elements are added to music of other genres, the indicators of Classical instrumentation are overpowered. For example, Septicflesh's music contains an obvious symphonic element, but the model was wholly unable to detect it. Other genres that are easily drowned out are: Doom Metal, whose presence was not only unidentifiable but also obfuscated other constituents (mixed-genre songs that contained Doom Metal had an average score of -0.549); Industrial Metal, where even pure-genre music varies wildly between artists; and Punk Rock, which is only represented in mixed-genre form by later Darkthrone, which is a far cry from the pure Punk Rock the model was trained on.

Polynomial SVM

As in the case of the linear model, backward elimination yields a reduction of three predictors. However, in this case the pre-tuned accuracy is far lower (0.18 vs. 0.425). This is likely because polynomial SVM's require three parameters to be tuned whereas linear SVM's require two.

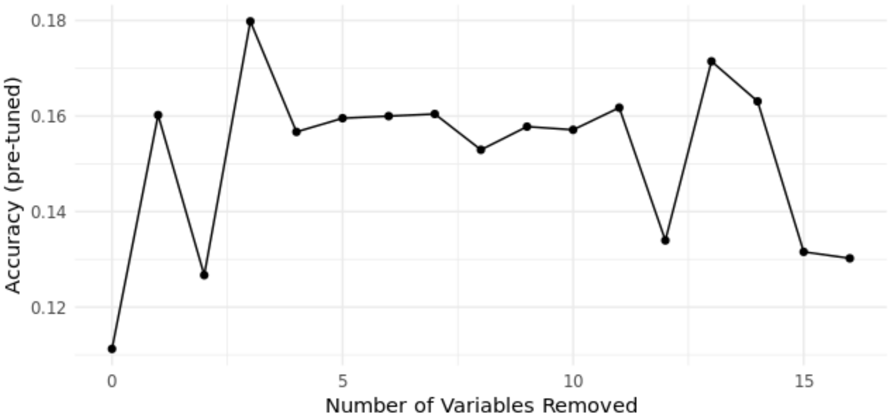


Figure 8: Polynomial SVM - Backward Elimination

After tuning and training, this model classified pure-genre songs **50.0%** correctly, a 1.6% drop from the linear model but still much higher than would be expected from random selection. Given the non-random sampling of the music population that this dataset represents, it is difficult to determine whether this slight decrease is significant.

The success of the model across genres is shown in Figure 9.1. As with the results of the linear model, Classical, Death Metal, and Electronic take the top spots, but there is a bit of a shake-up among those that follow. Power Metal and Melodic Death Metal took a tumble from approximately 50% success in the previous section to barely being distinguished at all, and Black Metal jumped from approximately 35% to 75%. The general shape of the distribution of success has shifted: the top 5 genres all exceed 50%, but, from there, there is a steep drop off, whereas the linear plot is more gradual in its decline across the gamut of genres. This is intuitive because, in order to maintain a similar mean success rate, gains in the success of one genre must be balanced by losses in that of another (and these losses become more substantial if the benefiting genres are also the most numerous in the sample).

Figure 9.2 depicts the same genres with their respective sample sizes. A similar relationship to the previous model is obtained (positive correlation between sample size and success), however it is concerning that Power Metal is relatively numerous but is almost complete unrecognized by the model.

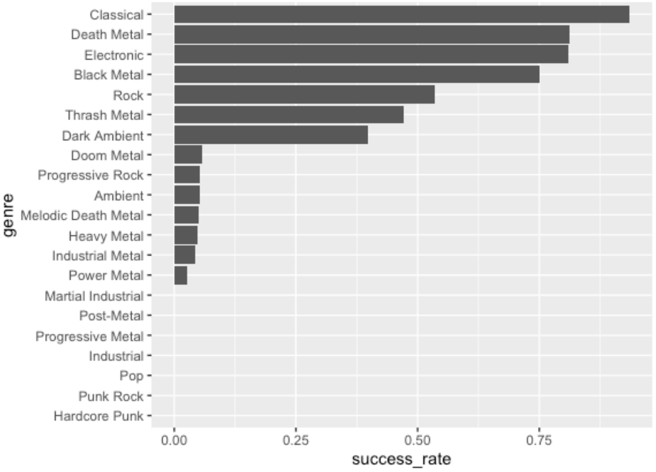


Figure 9.1: Polynomial SVM - Preliminary Success Rates

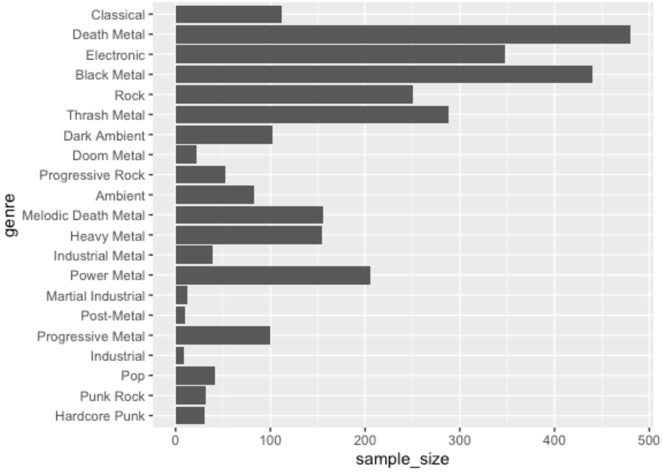


Figure 9.2: Polynomial SVM - Training Set Sample Sizes

This is potential evidence of a fundamental error in the polynomial model: Power Metal is easily distinguishable by anyone familiar with the metal scene, but, even with a sizable training set, the model cannot identify it. Further evidence of underlying problems lies deeper. One could justifiably assume, based sonic similarities, that the model is grouping Power, Heavy, and Progressive Metal together and then confusing them in its predictions. However, this is not the case: the most numerous predictions of Heavy and Power Metal are Death and Black Metal. Even a novice listener could tell you that these genres are not at all similar as far as metal sub-genres go (pure Power Metal vs. pure Death Metal). Despite the fact that class weights were provided in the creation of the SVM, there still seems to be a good deal of naïve classification: Black and Death Metal are the most well-represented metal sub-genres in the training set and the model seems to be disregarding the rest so as to perfect its classification of these three. Needless to say, this is not ideal.

After adjusting the model to consider multiple predictions, the resulting average score is **-0.0103**. This is slightly higher than the average score for the linear model but not by a significant margin. Therefore, it may seem that they are equally useful. However, as with the pure-genre songs, the distribution of success across genre varies and there are underlying problems.

Figure 10 depicts the successful detection rate for each genre in the full test set and in the mixed-genre portion only. As with the linear model, the success of Classical plummets once it becomes mixed with other genres: here, it falls from nearly perfect detection with pure-genre songs to zero with mixed-genre songs. Unlike with the linear model where Melodic Death and Death Metal were clearly distinguished through both stages, here, the success of the former is clearly riding on the greater success of the latter. For instance, only 13% of the Melodic Death Metal tracks (pure and mixed) in the test set were primarily predicted as such; 75% were primarily predicted as Death Metal. Therefore, Melodic Death Metal's increase in successful detection from approximately 5% in the preliminary polynomial stage to over 50% here is a function of it being predicted correctly at the secondary or tertiary level. Contrary to the results of the linear model, this is evidence that the two genres are not distinct but that one stems from the other.

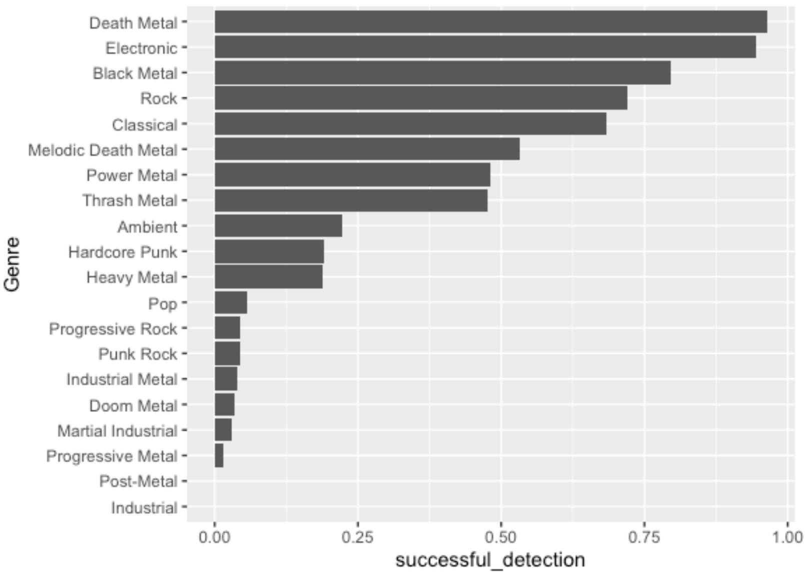


Figure 10.1: Polynomial SVM - Full Test Set

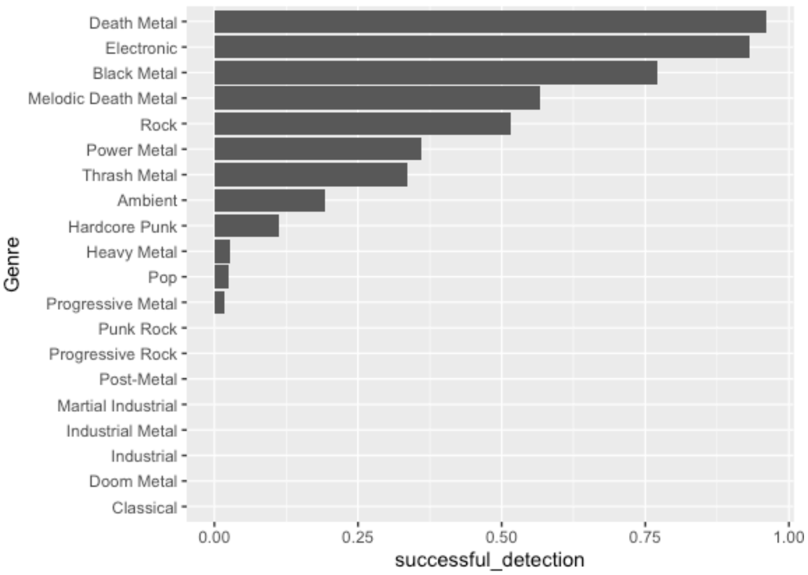


Figure 10.2: Polynomial SVM - Mixed-Genre Songs

However, the strength of this evidence is diminished by an analysis of Power Metal's also sizable increase in successful detection. As with the pure-genre set, the model is conflating Power Metal and Death Metal: only 5% of the former are primarily predicted correctly, whereas 73% are primarily predicted as the latter. It is promising that Power Metal is at least becoming more successful in the secondary predictions, but clearly the more numerous genres have their hands on the scales so to speak.

The naivety of this model is demonstrated by Figure 11.1, which depicts the false positive rate for each genre. Over 50% of songs in the test set were misclassified as Death Metal. Clearly, the sample size and/or sonic distinctiveness of the genre was enough for the model to disregard many of the rest. A similar, but less severe, phenomenon takes place for Black Metal.

For comparison, Figure 11.2 illustrates the same false positive rates for the previous linear SVM. Notice how even the worst offenders barely break 20%. Now, it is possible that the polynomial model's ails could be remedied with a much larger training set, but that can not be determined without using one. With this in mind, and given that their average scores are negligibly different, the linear model should be preferred going forward.

Radial SVM

Given that the linear SVM outperformed the polynomial SVM, it is not likely that a radial model would yield any improvements. And so it is. The preliminary radial model classified only **12.5%** of pure-genre songs correctly. This is far enough below the approximately 50% classification success of the previous two models that the radial model need not be pursued further.

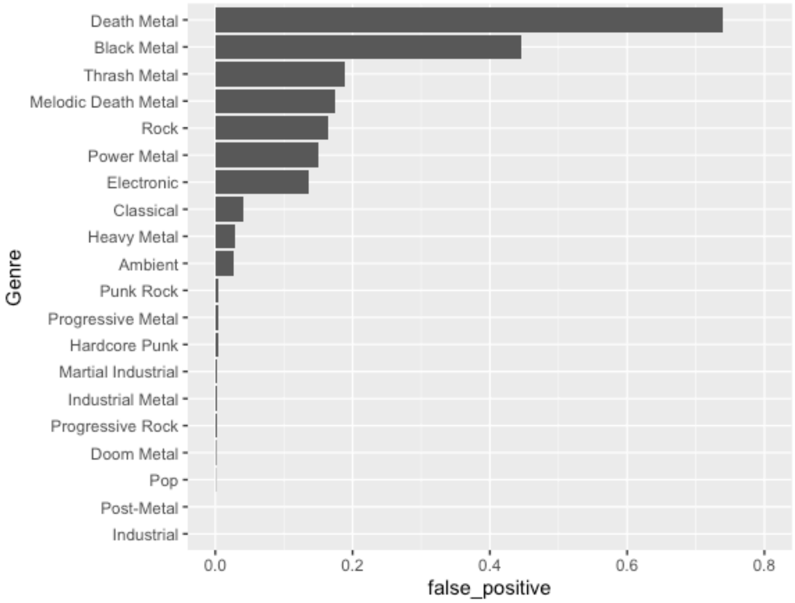


Figure 11.1: Polynomial SVM - False Positive Rates

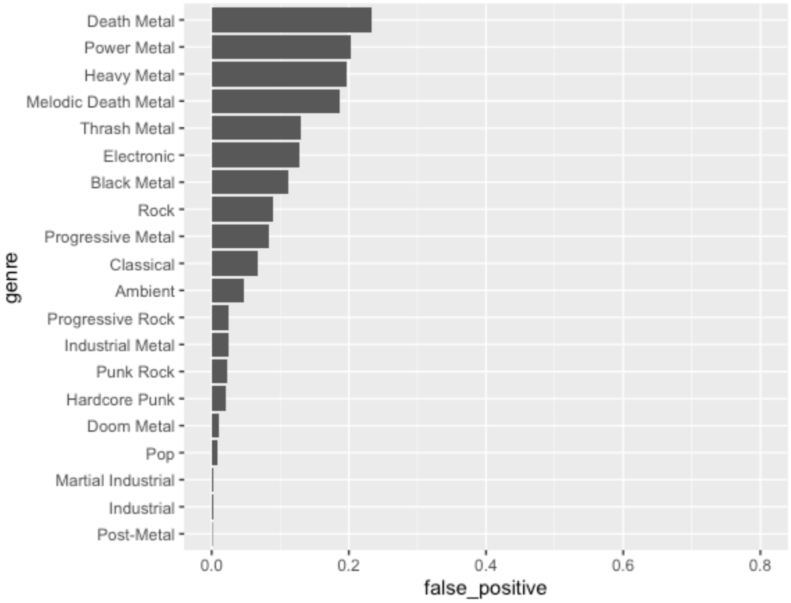


Figure 11.2: Linear SVM - False Positive Rates

Potential Developments for the SVM Method

A potentially promising extension of the SVM-based classification method is the construction of a decision tree incorporating SVM's at each non-terminal node. This structure is easy to conceptualize as most people intuitively classify music in this manner: hence the terms genre and sub-genre.

Classification begins at macro-level genres or macro-level trait decisions. Following the results from the first prediction, the observation progresses to a new classification node along the appropriate path. This process continues until the observation arrives at a terminal node and is classified as the high-resolution sub-genre that resides there. For example, consider Figure 12 (a simplified tree) and a Progressive Metal track. If this observation is classified correctly, it will go through four stages: 1) does the track contain percussion? Yes, 2) the track is Rock, 3) the track is Heavy Metal, and 4) the track is Progressive Metal.

Because the decision tree is an algorithm and not necessarily a correct musical map, identical terminal nodes can exist at different points on the tree. For example, here, Ambient appears appears under "Percussion? No" and under Electronic because some Ambient music contains percussion and some doesn't. This means that the example Progressive Metal track could also be correctly classified if it were to pass through Progressive Rock.

This process could result in greatly improved accuracy by reducing the number of categories for each SVM. The previously analyzed SVM's considered a set of approximately 20 genres that were as specific as I could manage. A decision tree could cut this down substantially. In the demonstration tree, the first SVM would only consider two classes: those with percussion and those without. The SVM at the

"Percussion? Yes" node would only have to consider five classes: Classical, Rock, Electronic, Jazz, and Folk. And so on until a terminus is reached.

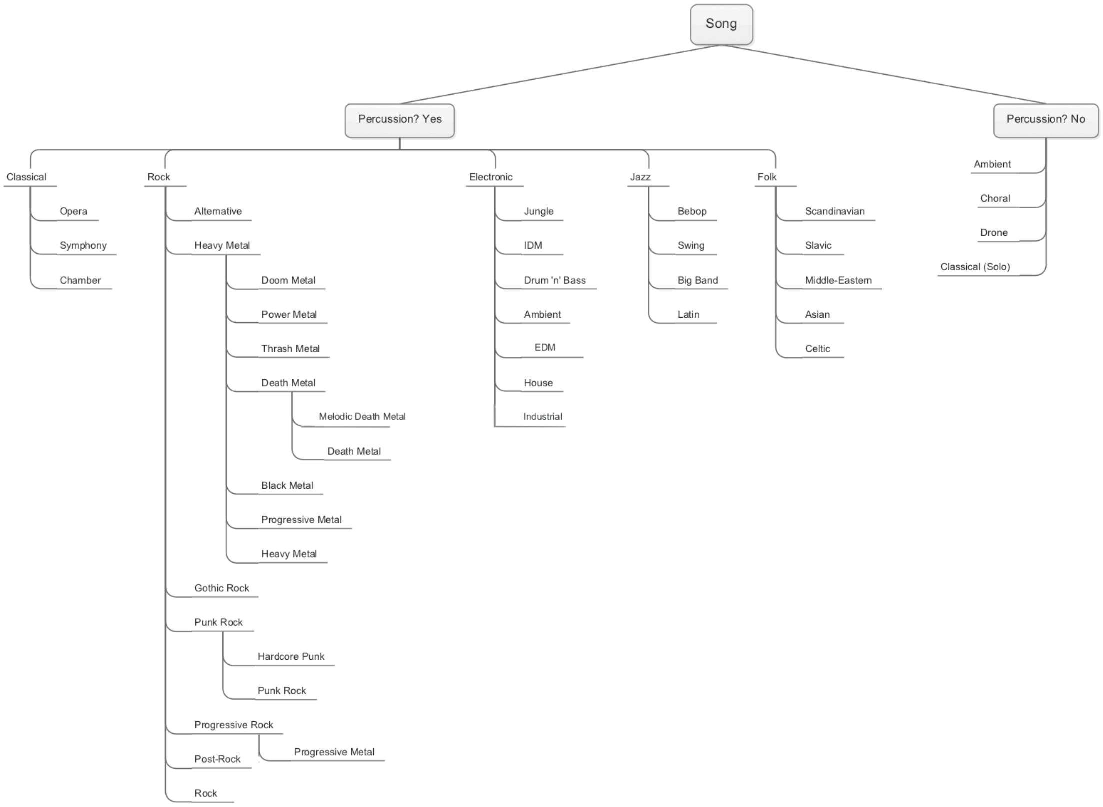


Figure 12: Simplified Decision Tree

As before, the model would classify each frame and then process all predictions to put out a classification for the whole track.

A second, much more complicated, development involves altering the prediction mechanism of the SVM's themselves. Up to this point, an assumption has been made as to the structure of mixed-genre music: for each track, there are x frames that are clearly Genre 1, y frames that are clearly Genre 2, etc.; generally, assuming that mixed-genre music is comprised of differing pure-genre frames. While this is certainly true for some songs, most are mixed more evenly, in that even the constituent elements of the tracks are mixed. For example, consider a song whose frames are consistently 75% Genre 1 and 25% Genre 2. A perfectly accurate version all the models considered so far would classify each frame (and thus the whole song) as purely Genre 1; a superior model would classify each frame as both genres.

An SVM with this capability must calculate the distances from the values of each frame to the multiple hyperplanes that constitute the decision space. As a demonstration, Figure 13 depicts a two-dimensional SVM with two possible classes: Genre 1 and Genre 2. The dotted lines are not the support

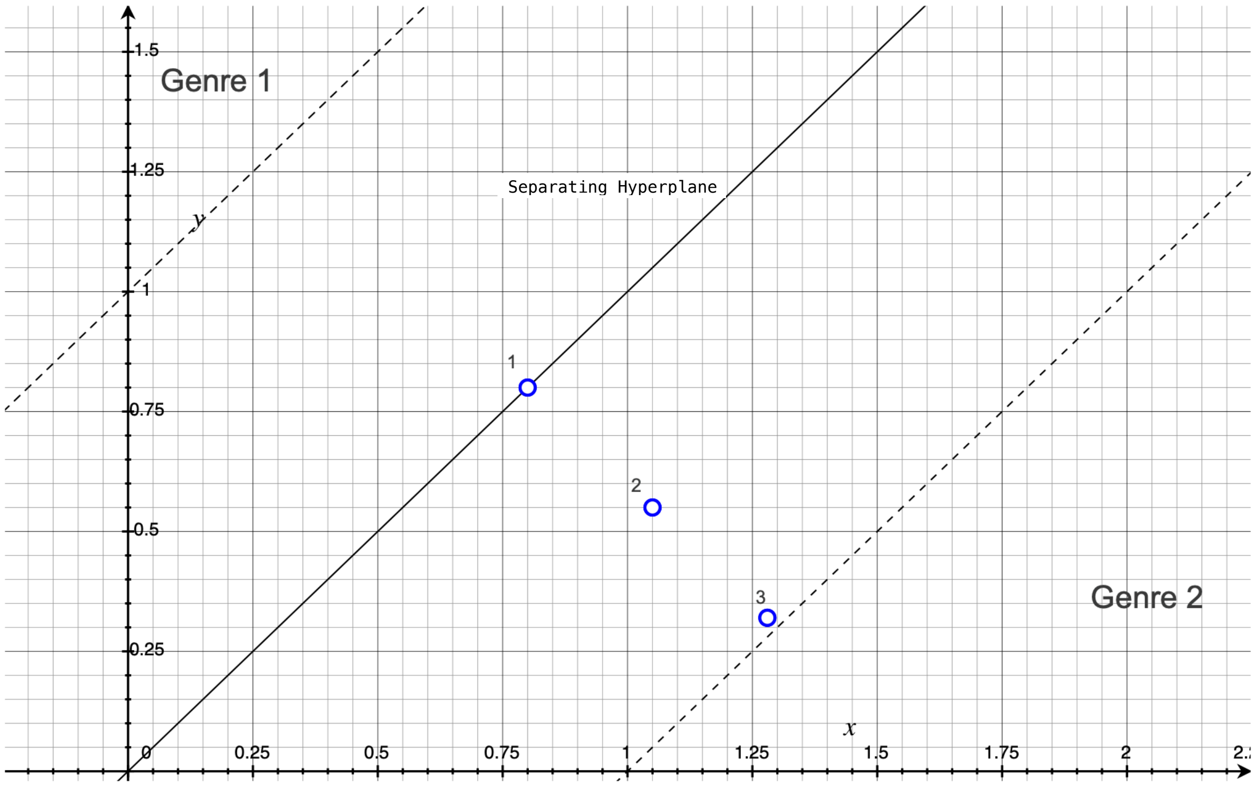


Figure 13: Two-Dimensional SVM

vector margins but are the borders of pure classification: beyond these lines, observations are classified with a singular prediction; between them, predictions are mixed. For example, consider points 1, 2, and 3. Point 1 should be classified as 50% Genre 1 and 50% Genre 2, Point 2 should be classified as 25% and 75%, and Point 3 should be classified as 1% and 99%.

Creating this advancement would be quite difficult because the SVM functions in the "e1071" package that were used in this project would have to be re-written to put out the coordinates of the various hyperplanes in *p*-dimensional space, and then numerous additional parameters would need to be tested to establish the positions of the dotted hyperplanes. This would also take a substantially larger dataset, with observations that represent substantially more genre-combinations. This could potentially be conducted by a commercial music provider with more data, more computing power, and more engineering wherewithal, but, since I currently lack all three, this advanced method must be tabled for now.

Graphical Mapping

Even if both of these developments were implemented, there remains one insurmountable obstacle for all of these models: the interactions between frames are completely lost, and the distinctions between genres may only be visible across chronological frames. For example, to the human ear, many similar genres are distinguishable by different rhythmic patterns: this is obvious when contrasting Speed

Metal with its successor Thrash Metal. If spectrometry information is plotted in chronological order, something like this can be seen in the magnitudes and frequency of the peaks and valleys. Because this information is lost when frames are disconnected, it is likely that the success of any frame-classifying algorithm is largely due to different combinations of instrumentation and sonic-tone/production-value being highly correlated with specific genres.

However, even a trait as important as instrumentation does not always map perfectly onto genre. For example, the standard instrumentation for Rock Music is a drum set, a bass guitar, an electric guitar, and vocals. While a detection of this setup could provide some intuition about the music, the correct classification could still fall into any number of genres, from standard Rock ’n’ Roll to Hardcore Punk to Death Metal. Additionally, this set up doesn’t provide any information about song structure or composition, which are, in reality, the defining measures of genre. For example, the only real similarity that Post-Rock music shares with Rock writ large is the instrumentation itself.

This point becomes more obvious in the context of synthesizer music because just about any genre of music can be constructed with a synth. For example, Norrin Radd’s *Anomaly* uses a chipset to create Death Metal music; therefore, the instrumentation falls under the Electronic umbrella term, but the genre is Death Metal. As one would expect, the linear SVM outlined previously classified these tracks as purely Electronic (the polynomial model correctly classified a few of them, but that is likely only because of the naïve predicting previously discussed). While cases like this, where there is a wide gap between the genre(s) typical of the instrumentation and the actual genre of the composition, are not common, the fact that it can happen at all means that instrumentation itself is not necessary nor sufficient to classify a song as a corresponding genre.

Needless to say, an algorithm that parses simply for instrumentation cannot possibly be ideal, although it may perform well generally. And this is likely the case for the SVM’s in this project and throughout the current literature.

To bridge this gap, I have created a modified K Nearest Neighbors (KNN) model which I have termed “graphical mapping” (the name should become obvious shortly). It is far too computationally intensive for me to test on a large scale, but I believe it holds the most promise of any model explored in this project.

KNN is a relatively simple but very powerful method of classification. In this method, a single test point is compared to an existing dataset of reference points with no intermediary modeling structure. All points are placed in p-dimensional space (where p is the number of predictors), and the Euclidean distances between each reference point and the test point are calculated. The reference points are then ordered by distance to create a list of “nearest neighbors”. When $K = 1$, the test point is predicted as the classification of the nearest neighboring point. As K increases, the prediction is the most frequent classification of the top K nearest neighbors (weighted schemes notwithstanding).

Here, we will be considering a five second interval as the test point. In order to compare this interval with the reference set, each reference point (song) must be decomposed into $l - s$ intervals, where l is the number of frames in the song and s is the number of frames in a five-second interval.

If one had the computational power to fully test this method, the following aspects must be tuned:

1. K: the number of neighbors considered
2. weights for each neighbor position
3. how to handle mixed-genre neighbors
4. when to put out a mixed-genre prediction
5. weights for all 62 variables as it is highly unlikely that all are equally important for prediction.

Additionally, multiple intervals from the test song would need to be evaluated to avoid the selection of non-representative segments.

To demonstrate this method, the full 40 Gb dataset must be used because the reduced set was a modified random sample (which eliminates chronology). We will be classifying Kimbra's "Carolina" using only the "centroid" predictor (as my computer cannot handle the full set of predictors). The reference set will be all other songs not by this artist.

Figure 14 shows the centroid values for frames 5000 to 5216 (five seconds) of "Carolina" plotted against the centroid values of frames 1 to 217 of Iron Maiden's "Run to the Hills" (by all accounts, a song of very different genre). For all 216 points, the distances are calculated between corresponding points and then averaged to generate a measure of difference for that interval. This process is repeated for every 216-frame interval of "Run to the Hills" (and so on for all other reference songs). The second plot in Figure 14 shows the 100th interval of "Run to the Hills" being compared to the same test interval.

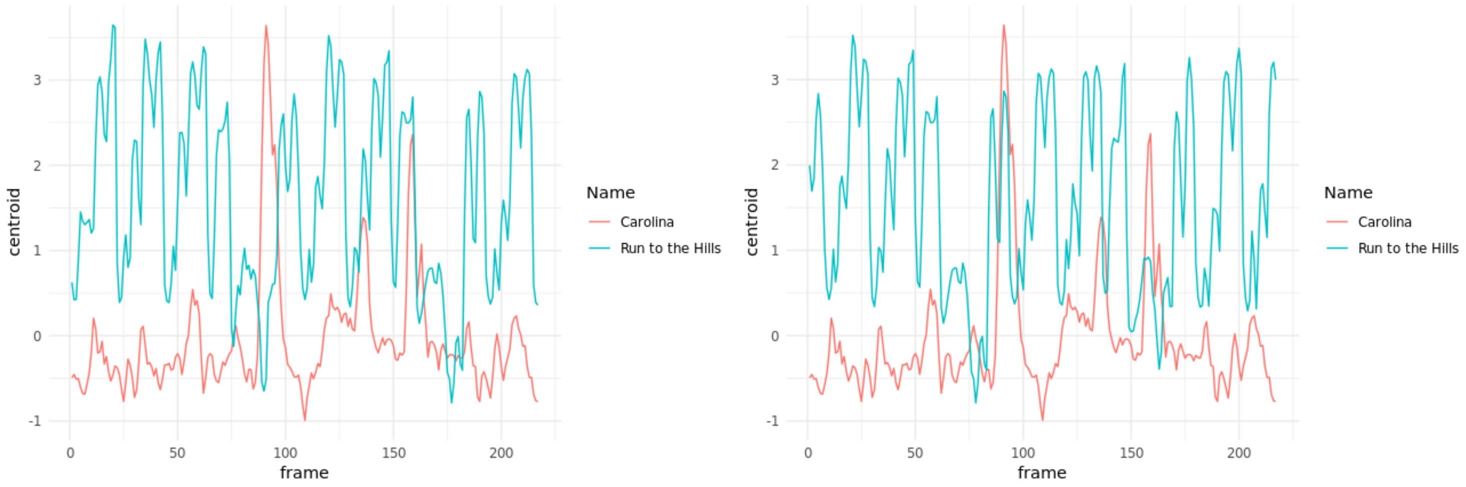


Figure 14: "Carolina" Contrasted with "Run to the Hills"

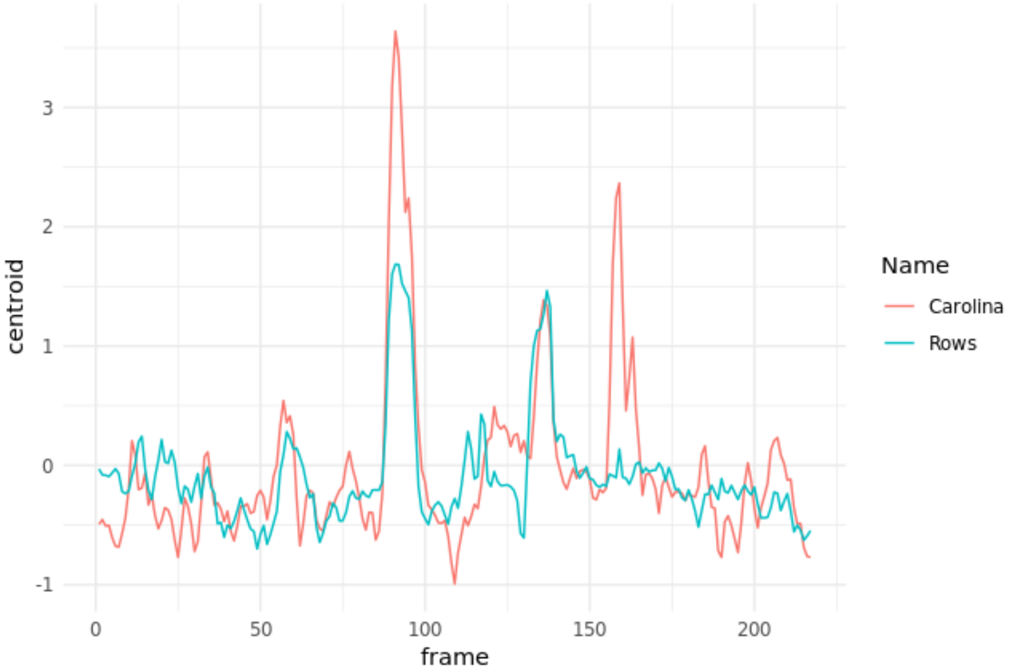


Figure 15: “Carolina” Classified by “Rows”

When this process is completed for all reference songs, the reference intervals are ordered by their difference measures to create a list of nearest neighbors. Figure 15 depicts the nearest neighbor for “Carolina”: an interval from Mew's “Rows”. Despite the fact that the Kimbra track is Art Pop and the Mew track is Indie/Alt Rock, the two tracks are relatively similar at certain points (specifically, in the latter parts of “Rows”), meaning that this preliminary assessment of the method, using only 1 out of 62 potential variables and a relatively small dataset, is quite promising. In fact, given the other music in the reference set, I believe this prediction was as successful as could be.

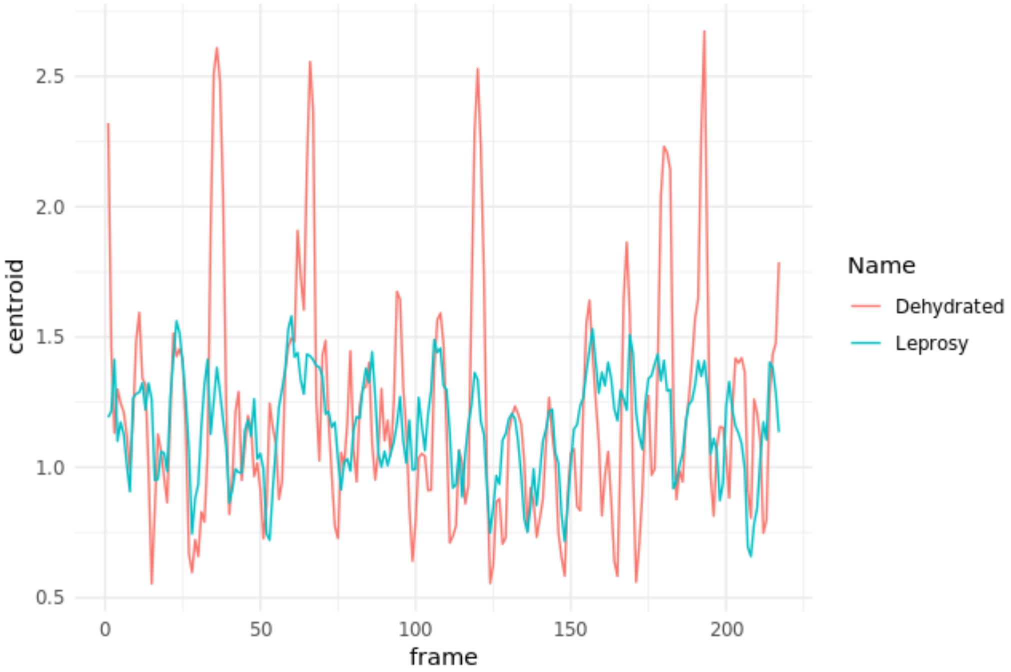


Figure 16: “Dehydrated” Classified by “Leprosy”

As another example, the final figure depicts the results for classifying Pestilence’s “Dehydrated” with the same method: it is correctly classified by the nearest neighbor, Death’s “Leprosy”.

While these results bode well, I have no way to test the model rigorously. Classifying a single interval with a reference dataset of 6123 songs with only one predictor takes most of a day to complete. (I’ve tested about 9 pure-genre songs with the centroid feature: 5 were predicted correctly, 2 were predicted incorrectly, and 2 were matched with songs of different genres but similar sounds like in the first example.) Therefore, considering the other 61 variables and tuning the necessary parameters would take an absurd amount of time.

While I hope to one day re-run this project with a much larger dataset and the more advanced SVM models outlined previously, testing the Graphical Mapping method will be have to be left to academics or professionals in the music industry. I believe it has significant potential, and, if in the future processing power becomes a non-issue, I think it could be deployed commercially by the major streaming services.