Music Genre Classification through Simple and Echonest Meta Data Features using the FMA Dataset

Catherine Baker
Computer Science & Mathematics BS
Computer Science MS Expected
Emory University
Atlanta, GA, 30322, USA
cwbake2@emory.edu

Thomas Davidson
Computer Science MEng, BA
Computer Science PhD Expected
Emory University
Atlanta, GA, 30322, USA
thomas.james.davidson@emory.edu

Abstract

Music genre classification is a key task for music distribution and streaming companies to successfully recommend music to their users. Prior work in the field has focused on genre classification through the processing of raw audio signals and text-based or lyrical data. This processing requires high computational and storage expense as compared to analysing metadata instead. There has been very limited work examining the use of metadata as a way of predicting genre and we provide a preliminary basis for understanding how metadata, and more specifically 'simple' metadata, can affect reported accuracy in music genre classification. Our results provide promising avenues for saving in both storage and computation costs by using metadata features instead of relying on expensive audio signal processing techniques.

1 Introduction

Even before electricity, music has been a facet of culture that brings people together. The rise of the internet, web applications, and mobile applications have only made sharing music easier and easier. Nowadays, companies like Spotify and Pandora make millions of dollars in profit off of their extensive music libraries and services. For these companies, tasks that classify, recommend, and share music are incredibly important, and making the approaches to these tasks more computationally efficient and accurate is a key concern. More specifically, the task of music genre classification is at the forefront of this research.

1.1 Music Genre Classification

Music Genre Classification is a machine learning task which takes information about a musical track as the input and outputs an associated genre label. Being able to accurately predict a piece of music's genre can help companies make better recommendations to users based on current and past music interests.

1.1.1 Single vs Multi-Genre

Genre classification can be divided into two groups: single genre classification and multi-genre classification. Single genre classifies music into a single ultimate genre. Using combined models and features, this often is implemented as majority voting classification tasks. Multi-genre classification provides the top few predicted genres for a single piece of music. This classification technique allows for more versatile recommendations in music, but can be more expensive in memory and computation.

1.1.2 Audio Signal vs Text-Based vs Metadata

Music classification tasks change based on what features of the musical track are analysed. Most music genre classification techniques use the content of the music to classify the piece without regard to the lyrics or metadata (Corrêa and Rodrigues, 2016) (Oramas et al., 2018). This most often comes in the form of audio signals, which can contain a plethora of features for use in classification. The use of audio signals has seen accuracies for classification reach around 80% and models often compare the size and length of audio signals as well as pull lyrical data through speech recognition (Li et al., 2003). However, using audio signals comes with large storage and computational expenses when compared to simple numerical features or text-based data.

Many of these audio signals have been broken down into more quantifiable pieces of data, like height, length, or period of the audio waves, or more human interpretable features like acousticness, valence, danceability, energy, etc, and stored in publicly available datasets. This data is able to be stored much more cheaply, and can be processed and compared to other data more quickly (Satriya Rahardwika et al., 2020). Many datasets bolster this extracted audio signal data with more information surrounding the lyrics and simple metadata (e.g. artist name, track name, date or location recorded). This allows researchers to pick and choose which features they want to include in their approaches based on a large set of musical tracks.

1.2 Music Datasets

However, metadata extraction from audio signals, lyrics, and musical pieces in general is still a fairly new approach to music genre classification research despite it being used internally at companies like Spotify for many years(Hern, 2014). There are still many research tasks based solely on finding better ways to extract this metadata from musical pieces. Most work to date extracting metadata from audio signals has focused on speech recognition and music-speech discrimination features like lyrics, whilst much less work has been done on descriptive features of audio signals like acousticness or energy (Li et al., 2003). This leads to holes in many datasets in their metadata features, making it difficult to accurately classify genre as the sample size of datasets may decrease with feature inclusion (Satriya Rahardwika et al., 2020), a problem we discuss in depth in Section 5.1.1.

2 Related Work

Much research has been done into music genre classification utilising audio and occasionally lyric data. This data is most widely available and most extensively pre-processed. The majority of work focuses purely on using audio signal data (Ajoodha et al., 2015), though some work has combined this data through multi-modal approaches with Neural Networks (NNs) (Oramas et al., 2018), and other research has also included text-based data such as music reviews (Oramas et al., 2016a). A very small handful of papers have attempted to work with metadata and text-based data whilst not using audio data (Ignatius Moses Setiadi et al., 2020; Satriya Rahardwika et al., 2020). Below we will detail some of the nuances of these different approaches.

2.1 Approaches Using Raw Audio Signals

Both Oramas et al. (2018) and Kostrzewa et al. (2021) process audio signal data for their approaches. Oramas et al. (2018) uses a multi-modal approach, combining the raw audio signal data with lyric data. Kostrzewa et al. (2021) uses a single model approach, analyzing only the raw audio data. Both use NNs and extensive data sets for their research. Our approach builds from this work but instead incorporates analyses of the metadata associated with these audio signals rather than the signals themselves. We hope we can reach similar accuracy standards at less storage and computational expense by relying only on this metadata.

2.1.1 Multi-Modal Approach

Oramas et al. (2018) performs multi-modal deep learning using the MagnaTagATune (MTAT)(Law and Von Ahn, 2009) and Million Songs Dataset (MSD) (Bertin-Mahieux et al., 2011). The MTAT dataset stores audio signals from musical tracks and associates them with tags related to mood, instruments, and music style. The MSD provides a collection of 1 million songs with audio features and simple metadata on each track. Oramas et al. (2018) uses the raw audio signal, lyric data, MTAT tags, and some simple metadata to classify each track by genre. The authors use Convolutional Neural Networks (CNNs) to extract features from the raw audio signals and Recurrent Neural Networks (RNNs) to process some of the lyric data which requires sequential context to be understood. This approach allows the authors to conclude that multi-modal approaches to music genre classification can far outperform any single modal approach.

2.1.2 Single Modal Approach

The paper by Kostrzewa et al. (2021) uses a small subset of 8,000 tracks from the Free Music Archive (FMA)(Defferrard et al., 2016). This subset is a collection of the top 8 genre labels from the FMA dataset each with 1,000 associated tracks. Each track contains a 30 second audio clip, various genre tags, and metadata. The authors used a CNN to extract features from the fixed-length audio signals. They then used 1-D and 2-D Convolutional Recurrent Neural Networks (CRNNs) and an RNN to extract the temporal dynamics of the audio signals. These models are then ensembled in various combinations to reach the same performance as state-of-the-art reported accuracies in music genre classification. This paper serves as a recommendation to ensembling multiple NN models to achieve better genre classification performance from audio signals.

2.2 Multi-Modal Approach Using Text-Based Data

The Multi-modal Album Reviews Dataset (MARD) (Oramas et al., 2016b) is enriched by Oramas et al. (2016a) with data from the MusicBrainz[1] and AcousticBrainz[2] databases, combining affective, semantic, acoustic, and metadata features on songs with their reviews on Amazon. The text-based Amazon reviews are then trained on a Naive Bayes model using a Bag-of-Words approach. The more acoustic features are trained on a Support Vector Machine (SVM). The authors then run various combinations of these models to find the highest genre classification accuracy. Oramas et al. (2016a) found that the incorporation of text-based semantic information in their models greatly improved accuracy. The authors also conclude, like Oramas et al. (2018), that a multi-modal approach to music genre classification holds more potential to achieving higher reported accuracies than any single modal approach.

2.3 Multi-Modal Approach Using Meta Data

One fear with music genre classification tasks is that some simple meta data will cause a 1-to-1 mapping of genre classification like artist name or album name. This is because artists tend to write or perform music that all falls into the same genre. So, when models are trained on datasets with features like album name or artist name included in the training and testing sets, models may learn to predict specific genres just by that given artist name or album name. While this can be helpful in genre classification, it is not helpful in predicting unknown genres for music that may come from unknown artists, and ultimately takes away credibility from these predictive approaches. We found 2 papers by Ignatius Moses Setiadi et al. (2020) and Satriya Rahardwika et al. (2020) that focus on a multi-modal approach using text-based and simple metadata and are not exempt from this issue in music genre classification.

Work by Ignatius Moses Setiadi et al. (2020) uses a subset of the Spotify Music Dataset (SMD) (Brost et al., 2019) with 6,000 tracks covering five to eight selected genres. This dataset includes metadata on audio features like speechiness, tempo, and valence, as well as more quantifiable aspects of the tracks like duration and time signature, and simple metadata like artist and track name. Ignatius Moses Setiadi et al. (2020) performed feature selection finding the top 13 features applicable to genre classification. These features were then passed to NB, KNN, and SVM models to perform genre classification. The authors achieved accuracies as high as 80% with SVM models.

Satriya Rahardwika et al. (2020)'s paper, by most of the same authors, uses the same subset of the SMD and the same feature selection process (resulting in the same features and the same use of artist name in the features). This time the authors used SVM only, tuning hyperparameters with k-fold cross validation. The authors again achieved an accuracy of 80% at its highest with SVM models.

The work by Ignatius Moses Setiadi et al. (2020) and Satriya Rahardwika et al. (2020) are impressive additions to the music genre classification field as they are some of the first researchers to use simple metadata in their selected features. However these authors both include artist name as a feature in training and test sets with no mention of an accounting for a one-to-one mapping of genres. Because of this, their results are dubious and may unfairly represent higher accuracies.

2.4 Novelty

Our research diverges from that of previous work by focusing explicitly on simple metadata like artist name, song duration, album name, and more. While this metadata is often used in music genre classification to label and identify tracks, to our knowledge it has only twice been used in genre prediction by Ignatius Moses Setiadi et al. (2020) and Satriya Rahardwika et al. (2020) and in these approaches was combined with additional extracted metadata related to the musical features of the track.

In this work we aim to investigate the feasibility of just using simple metadata to predict genre and provide a first exploration of which features may be of most value when predicting genre. We build on the the two papers mentioned here and expand this analysis to investigate how the extracted metadata about audio features, such as acousticness, can also be used to predict genre and compare and contrast how it performs compared to

[1] https://musicbrainz.org/doc/MusicBrainz_Database

[2] https://acousticbrainz.org

simple metadata. We compare our performances to this previous work and attempt to learn from and avoid potential pitfalls that might impact the validity of their results.

3 Methods

As recommended by Corrêa and Rodrigues (2016), Oramas et al. (2018), and Oramas et al. (2016a) we adopt both a single and multi-modal approach in our research analyzing text and numerically-based features on audio signal meta data and simple metadata. Following in line with Ignatius Moses Setiadi et al. (2020) and Satriya Rahardwika et al. (2020), we adopt the use of three simple classification models, mirroring the use of K Nearest Neighbours (KNN) and Naive Bayes (NB). Whilst we initially intended to use Support Vector Machine (SVM) model approaches, through our initial exploration it became clear that training these models would take prohibitively long even when working with very simple features and so took the decision to replace the SVM with a Stochastic Gradient Descent model (SGD) instead (a decision which we discuss further in Section 5). We first train and test single-feature models based on each individual feature. Then we will try to improve this performance through various combinations of features. Lastly we will attempt an ensemble approach, similarly to Kostrzewa et al. (2021), to predict genre through majority voting of these models.

3.1 Picking a Dataset

As we show in Section 2, there are dozens of possible datasets to choose from for music genre classification including, but not limited to, GTZAN (Tzanetakis and Cook, 2002), MSD (Bertin-Mahieux et al., 2011), MARD (Oramas et al., 2016b), FMA (Defferrard et al., 2016), and SMD (Brost et al., 2019) datasets. These datasets all provide different benefits and limitations.

GTZAN is the first dataset on music that was made publicly available. It contains 1000 clips of music from 10 different genres and some fairly simple data about them. While this dataset was a strong contender for our research, it comes with many recorded flaws such as mislabelling and distortion and contains very little actual metadata on the tracks. Despite this GTZAN remains one of the most popular datasets for music classification.

The MSD, discussed in Section 2.1.1 and used by Oramas et al. (2018), contains one million songs, most of which come with audio clips available via link. While this dataset is also very popular, it only contains very basic artist level metadata, which is ultimately not helpful to our research.

The FMA dataset is the newest of these and incorporates much richer metadata about the track. It is copyright free and makes musical tracks directly available to users. In addition, it also provides the extracted metadata from the Echonest[3] for a subset of around 15,000 songs, and simple meta data on a subset of around 56,976. Because of the breadth of richer and more helpful metadata along with having less data quality issues, we use the FMA dataset and various subsets of data within it to perform our analyses.

3.2 Feature Selection

We selected our features based on their availability within the dataset and their human interpretability. Track duration, track_listens, track_name, and artist_name are all covered (with no null values) on 100% of the songs in the dataset. Because of this we decided to include all of these simple metadata features in our training and testing sets. In addition to these four features, we also identified track_favorites (covering 61% of the dataset) and date_recorded (covering just 6% of the dataset) as preferable features to include. Because the sample size for the data subset including date_recorded is so small, we plan to include it separately from the other simple metadata features in our combined models. This way we will be able to address a possible bias within the data subset within our results.

3.3 Genre Selection

The FMA dataset contains 16 distinct genres of uneven distribution. Within a subset of the FMA dataset, there are 56,976 tracks labeled with a single top genre. We will refer to this subset as S_{TG} for clarity. The breakdown of how many tracks apply to each of these top genres for this subset is shown in Table 1. The top 6 tracks, Rock, Experimental, Electronic, Hip-Hop, Folk, and Pop make up for 86.38% of this data subset.

Table 2 shows the additional breakdown of the genre distribution within the subset of the FMA dataset including Echonest features (9,354 tracks).

[3] Now owned by Spotify.

Genre	% of dataset
Rock	28.59
Experimental	21.39
Electronic	18.89
Hip-Hop	7.16
Folk	5.65
Pop	4.70
Instrumental	4.19
International	2.80
Classical	2.48
Jazz	1.15
Old-Time / Historic	1.12
Spoken	0.85
Country	0.39
Soul-RnB	0.35
Blues	0.22
Easy Listening	0.22

Table 1: Genre distribution in the FMA dataset

We will refer to this subset as S_E for clarity. Here we can see the top 6 genres are Rock, Electronic, Hip-Hop, Folk, Old-Time/Historic, and Pop, accounting for 91.48% of the dataset.

Genre	% of dataset
Rock	41.61
Electronic	23.19
Hip-Hop	9.73
Folk	9.43
Old-Time/Historic	3.82
Pop	3.70
Classical	2.83
Jazz	2.58
International	1.42
Instrumental	0.9
Blues	0.71
Experimental	0.18

Table 2: Genre distribution in the subset of the FMA dataset containing meta data on echonest features

We intend to include six of the simple metadata features from S_{TG} in our work (track_duration, track_listens, track_favorites, track_date_recorded, track_name, and artist_name). We intend to include eight of the Echonest metadata features from S_E in our work (acousticness, danceability, energy, instrumentalness, liveness, speechiness, tempo, and valence). To make the task of classification less complex, we will account for the top five genres common to both data subsets (Rock, Electronic, Hip-Hop, Folk, and Pop) when running our models. This also allows us to more directly compare our performance with previous work in this area (Ignatius Moses Setiadi et al., 2020).

3.4 Pre-Processing

Our chosen features were organised into many different data types and representations. Firstly we had to process the .csv files from their original format into a usable dataframe. Many of the feature values required conversion prior to training and testing. Additionally, the labels for the features initially had many corrupted values. Some pre-processing was required to re-name these features for easier comprehension.

3.4.1 Numerical Conversions

Next we converted some features to more quantifiable representations. We first converted the 16 genres to numbered labels 1 through 16. This made genre comparisons in accuracy predictions easier and faster.

We also altered the values for date_recorded. The values started as datetimes in the form MM/DD/YYYY. To make them easier to plot and understand, we converted all datetimes to days_since_first, a value calculated by how many days a given date is from the minimum date in the set.

3.4.2 String Pre-Processing

Both artist_name and track_name were represented as strings. For each of these we processed and vectorised the string representations making use of the NLTK library. As part of the text processing we first shifted all the representations to lower case in an attempt to make the size of the data-space more manageable. Secondly, we experimented with the removal of stop-words and punctuation before using a Porter stemmer to again reduce the size of the dictionary that we would ultimately generate. Finally, we applied two different, common, approaches for vectorising: Term Frequency - Inverse Document Frequency (TF-IDF) and Bag of Words (BoW).

Through some initial experimenting we applied various different combinations of these processing and vectorisation approaches and report the results in Table 4. We found that a TF-IDF approach that removed stop words but did not remove punctuation (potentially reflecting the elevated importance and consideration of punctuation in song titles) produced the best results. Due to the time constraints of training models using song name (effectively equivalent to building a 28,000 feature model) we did not repeat these in-depth tests when our models became more complex and continue to work with this representation of song title when we include it as a feature in the rest of this paper.

3.4.3 Echonest Features

The Echonest features required significantly less pre-processing as the values were all normalised to float types between 0 and 1, except for tempo which was stored as its raw value. We applied a standard Sklearn MinMaxScaler[4] to the tempo feature as well as any other non-normalised numerical values.

3.4.4 Isolating Features

After all pre-processing was performed we isolated our chosen features into a separate dataframe and split them into test and training samples (75%/25%), ensuring this split was consistent for all model evaluation.

3.5 Single Feature Models

In the interests of building from the simplest and most interpretable models we first explored single feature models to understand how well each of our single chosen features could perform as predictors of genre. We ran each of our three chosen model types: Naive Bayes, Stochastic Gradient Descent, and K Nearest Neighbours on each of the 6 'simple' metadata features in Table 5 (which excludes artist_name) and each of the 8 Echonest metadata features in Table 6. These results are reported in Section 4.

3.6 Combined Feature Models

Having built and tested all of our single feature models we then proceeded to build more complex models representing linear combinations of features. We again split the reporting of our results from this exploration into combinations of 'simple' metadata features, Echonest features, and finally a combination of 'simple' *and* Echonest features.

When generating the linear combinations of features we did not include either track or artist name due to the impractical amount of time required to fit several thousand models to such a high dimensional dataset that would result from including the text representation. Instead we combined the best performing models from the combinations of other features with the track name to see if this improved the performance any further.

In total we tested the following number of different combinations of features:

- 'Simple' Feature Combinations - 11
- Echonest Feature Combinations - 247
- All Feature Combinations - 7,927

This resulted in a total of 8,185 different combinations of features that were tested and the full results can be found in our supplemental material in Section 6 whilst the best performing combinations are reported in Section 4.

3.7 Hyperparameter Tuning

Our goal for this project was an initial exploration of the problem space, so we attempted to keep the complexities of any models we created as simple as possible. To aid in this, we kept any parameters for the models we built equally as simple. This ultimately led to us using the default settings for most of the models that we generated, such as utilising hinge loss for the SGD model, and using the most popular GaussianNB model for our Naive Bayes model. We therefore did not do a huge amount of hyperparameter tuning as part of this project and left this to future work.

However, wherever possible we did our best to tune the value of K that we were using in our KNN models to ensure we were getting the best performance. This involved using Sklearn's grid search cross validation approach[5] which utilises a 5-fold cross validation approach for searching for the best K value over a specified range. We specified this range as typically being all odd values from 1 - $\sqrt{Number\_of\_Samples}$. In this way we selected an optimal K for our single feature and smaller combinations of features models.

All the optimal K's can be viewed in our supplemental material in Section 6. When incorporating

[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

the track_name feature, the models required a lot more time to process the large feature set that resulted from the text processing. Because of this, we searched over a the same range but at larger intervals for K in models incorporating track_name.

3.8 Boosted Models

Finally we turned to some ensemble methods to try to improve the overall accuracy of our models by combining the predictions from the different models we had built. Due to time constraints we were limited to building a simple VotingClassifier[6] which worked by combining the probability of predictions for each genre from the SGD, NB and KNN models using majority voting to select a genre. To ensure fair comparison we had to use 'hard' voting as the default hinge loss from the SGD did not allow us to utilise 'soft' voting and we did not have time to rerun all of our previous experiments with a different loss setting for the SGD.

4 Results

4.1 Dataset Size

As we briefly discussed in Section 3 we were unable to make use of the full 100,000+ tracks that are available in the FMA due to a combination of data quality issues, and our decision to limit the dataset to the top 5 genres across the echonest tracks and those tracks with a top_genre label.

As can be seen in 1 this resulted in the following 4 dataset sizes that we list here. Where possible we attempt to use the most relevant and largest subset for reporting the performance of the models and indicate this in our tables. In addition we include performances across all the datasets where possible for more complete comparison of different approaches to one another in our supplemental material in Section 6.

- S_{T5}: 32,241 tracks associated with the selected top 5 genres and having complete coverage of our chosen 4 'simple' metadata features

- S_{T5_D}: 2,711 tracks that are a subset of S_{T5} which also have a value for the feature date_recorded.

- S_{T5_E}: 8,192 tracks that are the intersection of S_E and S_{T5} that represents tracks that have Echonest features recorded and fall into the chosen top 5 genres.

- $S_{T5_{E,D}}$: 1,782 tracks that are the subset of S_{T5_E} that also have a value for the date_recorded feature.

4.2 Single Feature Models

4.2.1 artist_name

During the preliminary testing of some of our single feature models we discovered that we were achieving *very* high accuracy for the artist name as a single feature predictor for genre of a track, even as high as 88% using the KNN model. This is a surprising figure and one that did not make complete sense. After further investigation we ascertained that each artist was only being associated with a single genre. For example, every track in the dataset by the artist Ed Askew [7] was listed as having the top genre of 'Folk'. This points to potentially deeper problems with the way we categorise songs and artists into genres, but additionally it means that building a single feature model on artist name is deeply flawed because we are effectively building a one-to-one mapping of artist name to genre when we do this and when an artist appears multiple times in the dataset there is a high likelihood that in training the model will have seen multiple exact copies of the tuple (artist_name, top_genre) which it is then asked to evaluate as part of the testing. Effectively the models are just learning these associations and are not making any connections between the specific words used in artist names and how this might relate to genre.

To test this hypothesis we formed a new dataframe of just the artist names and their associated top genres, and removed all duplicates so that in training and testing a specific artist name would not be repeated. We re-ran our models and found the results much more in keeping with our other preliminary results (with the performance capping out at around 0.37 accuracy). The comparison of these two results can be seen in Table 3. [8]

As a result of these findings we decided to not include artist name as a feature in any of the further investigation or in the linear combinations of

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

[7] https://edaskew.bandcamp.com

[8] It should be noted that these results reflect trying to select the correct genre for the artist from our 16 genres as this testing was done before the decision to limit to a subset of 5 genres.

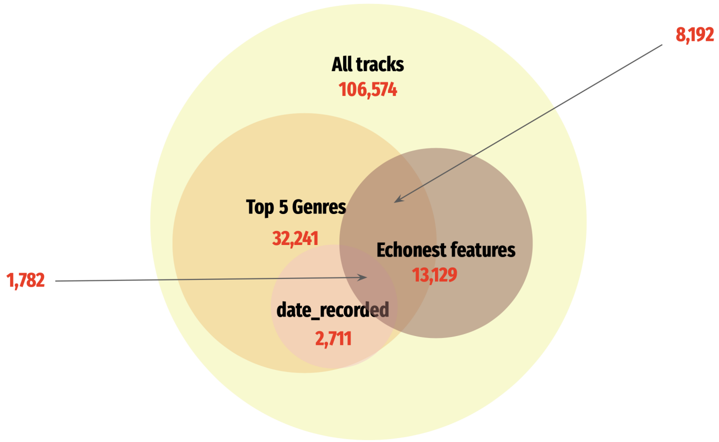

Figure 1: A Venn diagram showing the overlapping of our dataset. The 'Top 5 Genres' circle includes all data with a 'top_genre' feature label (S_{T5}). The 'Echonest features' circle includes all tracks with associated Echonest features (S_E) which overlaps with the top 5 genres circle (as S_{T5_E}) and the date_recorded circle (as $S_{T5_{E,D}}$).

	NB	SGD	KNN	Sample Size
artist_name_multiple	0.78	0.86	0.88	56,976
artist_name_single	0.2	0.37	0.34	9,070

Table 3: Best performance of NB, SGD, and KNN models on the artist_name feature including multiples of artist_name as artist_name_multiple, and excluding multiples as artist_name_single

features due to concerns of it having an undue impact.

4.2.2 'Simple' Metadata Features

Preparation of Track Name - As discussed in Section 3 we applied multiple different preprocessing and vectorisation approaches to the track name to find the best performing approach. The results of our initial experimentation (again with 16 genres) can be seen in Table 4. We found that using TF-IDF (instead of BoW) whilst removing stop words and *not* removing punctuation resulted in the best model performance. We proceed by using this pre-processing for all inclusions of track_name in feature models.

Table 5 shows the results of our single feature models for the 'simple' metadata features. The general performance of the numerical features are fairly comparable to one-another ranging from 43% to 46% accuracy across the models. The range of performance for track_name is much wider but we achieved the highest performance for any of the features covering the whole of S_{T5} with 50% accuracy for an SGD model using the track name to predict the genre. Whilst not particularly high accuracies these were very encouraging results to see for simple models built off individual features of the datasets and lends weight to our approach that leveraging even simple metadata can be an effective method for predicting genre.

We saw a significant increase in performance up to almost 80% for the days_since_first individual feature in a KNN model. This figure is suspiciously high for an individual feature and we tried our best to uncover exactly why the date a song was recorded was performing *so* strongly as a predictor for genre in Section 4.5.

4.2.3 Echonest Features

As a reminder, the Echonest features represent metadata that has been extracted by a black-box algorithm that reflects specific, human-interpretable, attributes of a track such as 'acousticness', or 'energy'. These features are utilised by companies such as Spotify (Hern, 2014) behind the scenes to drive their algorithms and recommendations.

As we can see in Table 6 the performance of the single feature models from the Echonest features is slightly better than the performance of the 'simple' metadata features. With a max performance of about 55% accuracy for the feature 'danceability'. Since the Echonest features should convey more information about the song than the simple metadata features this is not a completely surprising result.

We were concerned that the relatively smaller

track_name Processing	NB	SGD	KNN
TF-IDF - Include stopwords, remove punctuation	0.19	0.37	0.32
BoW - Include stopwords, remove punctuation	0.19	0.36	0.32
TF-IDF - Include stopwords, include punctuation	0.19	0.37	0.32
BoW - Include stopwords, include punctuation	0.19	0.36	0.32
TF-IDF - Remove stopwords, remove punctuation	0.19	0.37	0.34
BoW - Remove stopwords, remove punctuation	0.19	0.37	0.31
TF-IDF - Remove stopwords, include punctuation	0.19	0.37	0.34
BoW - Remove stopwords, include punctuation	0.19	0.37	0.31

Table 4: This table shows the pre-processing and vectorisation approaches for the 'track_name' feature. 'BoW' indicates that the Bag of Words approach was used and 'TF-IDF' indicated the Term Frequency-Index Document Frequency approach was used.

	NB	SGD	KNN
track_duration	0.43	0.44	0.46
track_listens	0.45	0.44	0.46
track_favorites	0.44	0.44	0.46
track_name**	0.25	0.5	0.44
days_since_first***	0.69	0.7	0.8
Average	0.3925	0.455	0.455

Table 5: Performance of NB, KNN, and SGD models on the single simple features from the FMA dataset. **Best pre-processing and vectorisation approach chosen. ***These results require more discussion and were excluded from our average calculation.

	NB	SGD	KNN
acousticness	0.47	0.47	0.46
danceability	0.55	0.48	0.55
energy	0.48	0.48	0.48
instrumentalness	0.47	0.47	0.47
liveness	0.47	0.47	0.47
speechiness	0.5	0.49	0.51
tempo	0.47	0.47	0.52
valence	0.47	0.47	0.47
Average	0.485	0.475	0.49125

Table 6: Performance of NB, SGD, and KNN models on various single Echonest features

	NB	SGD	KNN
track_duration	0.46 +3	0.47 +3	0.47 +1
track_listens	0.47 +2	0.47 +3	0.47 +1
track_favorites	0.47 +3	0.47 +3	0.49 +3
track_name	0.25	0.49 -1	0.49 +5
Average	0.4125	0.475	0.48

Table 7: Performance of NB, SGD, and KNN models on various single simple features within the data subset including Echonest features and excluding 'date_recorded' (S_{T5_E}). See Table 5 to compare results between datasets. The difference in value between these tables is given in green (for better performances) and red (for worse performances).

dataset size of S_{T5_E} compared to S_{T5} was potentially inflating our results and so we also trained models for the simple metadata features on the tracks that had Echonest features available and as can be seen in 7 there was no significant increase in the accuracies of these models and so we can lend more weight to the finding that **Echonest single features were better predictors than the single simple metadata feature models.**

Another interesting finding from the comparison of the single feature models can be seen in Figure 2 shows that although the maximum accuracy recorded from the single feature Echonest models was higher than the maximum accuracy for the simple metadata features, the range of values for these models was also wider. This is indicative that not all of the features from the Echonest were equally useful in predicting genre and some features were more informative than others - most notably, **danceability** and **speechiness.**

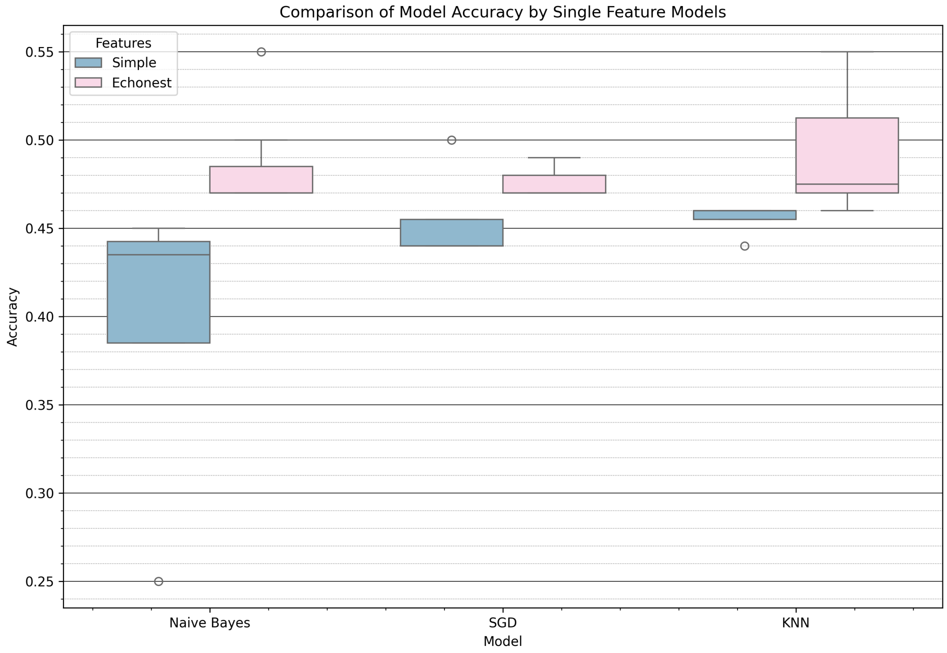

Figure 2: This box plot shows the distribution of our single feature models per feature type. In general the Echonest features showed slightly more data variance. The lack of whiskers for the simple features NB and KNN shows us that most reported accuracy values were very similar to each other. To see the full results per feature, see Table 5.

4.3 Combined Feature Models

We then proceeded to exhaustively search the data space to find the best linear combination of features for each of our three models: NB, SGD and KNN. We split this into three areas:

1. Combination of 'simple' metadata features
2. Combination of Echonest features
3. Combination of 'simple' and Echonest features

We present the results of the performances for these linear combinations of features in a set of annotated box and violin plots in an attempt to show the variation in performance and comparison of our combinations of features to the best performing single-feature models.

4.3.1 Combination of simple metadata features

As we can see in 3 we did achieve a very mild improvement in performance for certain linear combinations of features but there was no major increase in performance. The best performance was achieved by the SGD model which combined the 'track_duration' with 'track_listens' and the 'track_name' to achieve an accuracy of 0.52. We were only able to include the track name exhaustively for the NB and SGD models as including it for the KNN models took prohibitively long and did not seem to have a tangible impact on the performance - to see this please review the full results in Section 6.

When we included the track name as a feature for the Naive Bayes model we saw a drastic decrease in the performance of the model and this is likely due to the generally very low performance of the track name as a single feature for the Naive Bayes model (0.25) and the fact that any of the informative single features we did generate are now being drowned out as the track name represents several thousand new features (albeit the majority of these being 0 values) due to the vectorisation approach.

4.3.2 Combination of Echonest features

Figure 4 shows the distribution of performances for each of the combinations of the Echonest features. We see a noticeable increase in accuracy of these models compared to both the best performing single feature Echonest models, as well

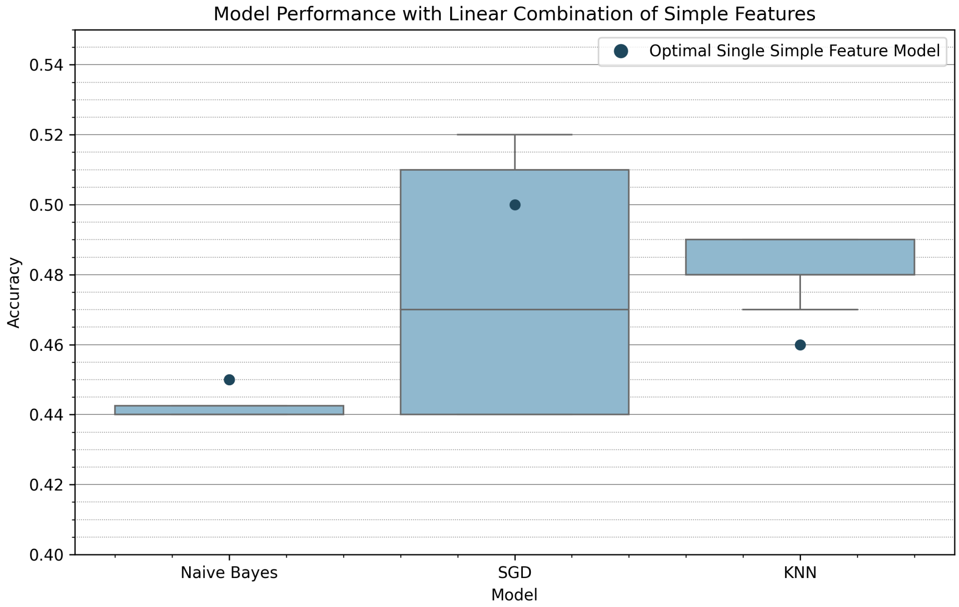

Figure 3: This box plot shows the distribution of accuracies for linear combinations of simple features. The dark blue dots represent the highest accuracy achieved by a single simple feature model (see Figure 2). For full accuracies per combination see Section 6.

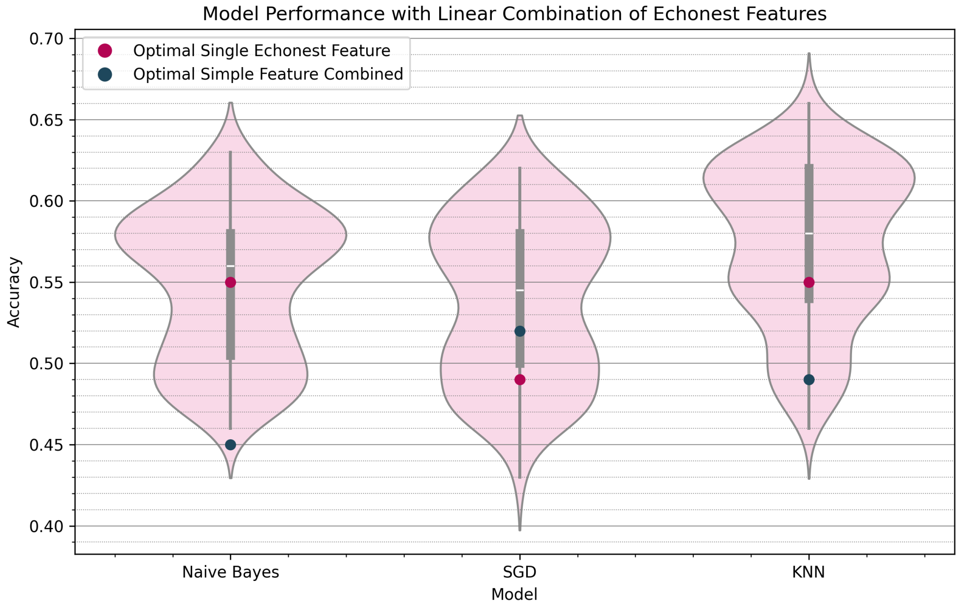

Figure 4: This violin graph shows the distribution of accuracies for linear combinations of Echonest features with dots added to show the highest performing accuracies for both single Echonest features (Figure 2) and combinations of Simple features (Figure 3)

as the linear combination of simple metadata features. We again see a range in performances implying that some of the features are not contributing as much to determining the genre as others and we see the highest performance of 0.66 accuracy from a KNN model consisting of all the Echonest features excluding 'liveness' and 'tempo' [9]. To see a summary of the feature combinations that make up the 20 best performing models please refer to Section 6.

4.3.3 Combination of all features

	NB	SGD	KNN
Simple	0.45	0.52	0.49
Echonest	0.63	0.62	0.66
All	0.62	0.63	0.66

Table 8: Best accuracy found through combination feature models using NB, SGD, and KNN with combinations of simple (S_{T5}), Echonest (S_E), and all features (S_{T5_E}).

Finally, in Table 8 we see that the addition of the simple metadata features to the Echonest features does not result in any noticeable increase in accuracy for the models - the biggest change being an increased maximum from 0.62 to 0.63 for the SGD model. This again implies that the Echonest features are more informative for determining genre. From initial investigation there was no particular simple metadata feature that when combined with the Echonest features resulted in noticeable increases in performance. The three numerical fields of duration, listens and favorites all appear fairly evenly across the top 9 performing combinations (read more in Section 6).

4.4 Boosted Models

As the final part of our attempt to improve the performance of our models we wanted to try and combine the output of the three different models we were using - SGD, NB, and KNN. We hoped that using a voting classifier we would be able to increase the performance of the different models and make them more robust. Our intuition was that some of the models were performing better for specific genres and so by taking a weighted average of the probabilities for each of the genres and then choosing the highest combined average probability we would reduce the number of misclassifications. However, as we can see in Table 10 we were unable to increase the performance of any of these combined models in comparison to the best performing individual models for both the single features or the combination of features. Unfortunately, the tight time constraints of this project meant we did not have sufficient time to investigate this and we outline some potential directions we intended to pursue in Section 5.1.

4.5 Limitations

Here we briefly outline some of the threats to the validity of our results and some other limitations that potentially hold us back from being able to draw stronger conclusions about our results.

4.5.1 Date Recorded as a Feature

As we discussed in Section 4.2 we saw very high accuracy reports for our calculated feature days_since_first which was a normalised float representation of the 'date_recorded' feature. We saw performance of values as high as 80% accuracy for this feature. We hypothesised that this could be due to some data quality issues where the date recorded was simply mapping directly to genres because all of a certain genre were being listed as recorded on a single date. However, we attempted to investigate this and plotted a chart to show the distribution of our different genre's recorded dates (Figure 5) and this does not seem to be the case.

More likely we find that the distribution of different genres is significantly skewed due to the severely diminished size of this dataset (2,711 records) meaning the model can achieve a fairly high accuracy by simply guessing one or two genres for every record. This hypothesis is likely to be the case when we consider the very similar results we see across the board for when we tested this dataset (S_{T5_D}) for our other single simple metadata features (Table 9).

	NB	SGD	KNN
track_duration	0.68	0.7	0.7
track_listens	0.71	0.7	0.69
track_favorites	0.69	0.7	0.7
track_name	0.34	0.66	0.71
days_since_first	0.69	0.7	0.8
Average	0.622	0.692	0.72

Table 9: Performance of NB, KNN, and SGD models on the single simple features from the FMA dataset including the 'date_recorded' feature (S_{T5_D})

[9] This is a linear combination of: acousticness, danceability, energy, instrumentalness, speechiness and valence

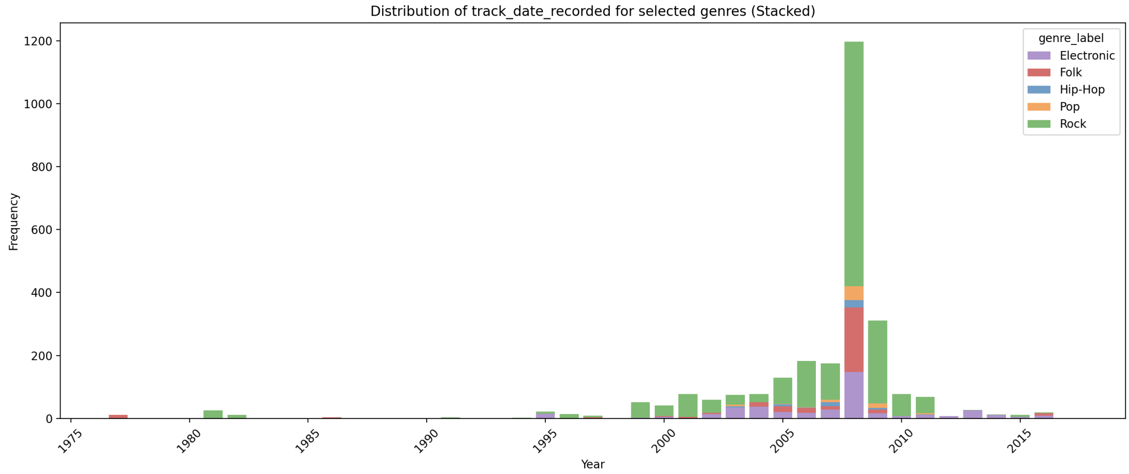

Figure 5: The date recorded values per genre. The color of the bars represent the genre those values apply to. The x-axis is by year to increase legibility and we were unable to identify any pattern at lower levels of granularity either.

Unfortunately we did not have time to further investigate this during our project and so instead we chose to focus on more reliable results. We outline ways we could address this problem in Section 5.1.

4.5.2 Low Number of 'Simple' Features

One of the key takeaways we hope to offer from this project is that the use of 'simple' metadata features is a viable option for use in predicting models. Our results show a lot of promise and being able to achieve accuracy of 52% (best performing combined features model) is a very promising result for a first approach at this problem utilising only 4 features. However, a key limitation of our project is simply not having enough metadata features or large enough dataset coverage to fully investigate how far performance of these simple features can be taken.

4.5.3 Performance of Models for Different Genres

One area that we wanted to investigate but due to time constraints were unable to do so, was the performance of our models on specific genres. In the end we chose to focus on reporting the overall classification accuracy as most prior work in this area does, but a fuller breakdown of understanding which genres were easier and harder to predict could have helped us to improve the robustness of our models as well as their overall performance.

4.6 Results Roundup

To summarise we were able to achieve the following best results for our 3 different classifier approaches of Naive Bayes, Stochastic Gradient Descent, and K Nearest Neighbours to classify the genre of musical tracks into one of 5 genres: Rock, Electronic, Hip-Hop, Folk, and Pop. Each of our optimal performances came from the combination of Echonest features, occasionally with the addition of one or more of our 'simple' metadata features. To see the full list of feature combinations please review Section 6. Our best results were **0.63** for a Naive Bayes classifier, **0.63** for a Stochastic Gradient Descent, and **0.66** for a K Nearest Neighbour's model.

Overall we saw that the 'simple' metadata features show some promise for allowing us to classify genre and this could be a very interesting area for future work. What is clear from Table 10 is that the Echonest features are better predictors of genre, both in their single feature models and in their combined models. In their current state we were unable to successfully apply ensemble methods to combine the models we built and improve their performance as classifiers.

5 Discussion

One of the goals of this project was to compare the performance we managed to achieve to that of the only other work we could find that attempted to achieve classification via metadata features rather than audio signal analysis. Ignatius Moses Setiadi et al. (2020) reported highest accuracies of 76%

	NB	SGD	KNN	Sample Size
Single Feature, Simple*	0.45	0.5	0.46	32,241
Single Feature, Echonest	0.55	0.49	0.55	8,192
Combined Simple Features	0.45	0.52	0.49	32,241
Combined Echonest Features	**<u>0.63</u>**	0.62	**<u>0.66</u>**	8,192
Combined All Features	0.62	**<u>0.63</u>**	**<u>0.66</u>**	8,192
Voting Clf Simple Combined	0.45			32,241
Voting Clf Echonest Combined	0.64			8,192
Voting Clf All Combined	0.64			8,192

Table 10: The highest reported accuracy over all single feature (*excluding 'date_recorded'), combined feature, and boosted models. Voting models report one accuracy because they use every model to vote on genre.

for NB, 77% for KNN, and 80% for SVM(Ignatius Moses Setiadi et al., 2020). In comparison we achieved 63% for NB, 66% for KNN and 63% for SGD (Table 10). Whilst our results do not quite achieve the same levels of accuracy as the Ignatius Moses Setiadi et al. (2020) paper as we touched on briefly in 2, the authors discuss utilising artist name as a feature whilst not discussing any attempt to mitigate the one-to-one mapping that we discussed in 4.2.1. In our own experiments we were able to achieve performance as high as 88% with a flawed approach for utilising artist name. Without further access to the methodology of the Ignatius Moses Setiadi et al. (2020) paper, it is difficult to draw firm conclusions about their approach and the possible false-confidence that their accuracy may have benefited from given this same issue.

In addition to this potential pitfall, the authors of this paper also had greater resources and were able to test and build more complex models such as their SVM approach which yielded their best results. Whilst we attempted to build SVM models for our work it ultimately was prohibitively time consuming and so we adapted to use a simpler SGD model instead.

The main contribution of this work is a thorough initial investigation into the use of metadata to predict genre. Our early results imply that **using metadata to predict genre *is* a feasible approach for music genre classification**. Moreover, we showed that it was possible to achieve some success using just 'simple' metadata such as track name and track duration although ultimately we found that preprocessed metadata such as the Echonest features allowed for better performing genre classification.

5.1 Future Directions

5.1.1 Need for Richer and More Complete Datasets

Whilst we were able to achieve greater than 50% accuracy for certain linear combinations of just the 'simple' features, we were severely hampered by a lack of rich and complete datasets. Although we identified and worked with the dataset which currently offers the most rich and best coverage of metadata features in FMA(Defferrard et al., 2016) we were still relying on building models with training sets of around 30,000 samples and having to rely on only 4 usable 'simple' fields. In order to attempt to utilise any of the other metadata features such as the date the track was recorded, we were forced to shrink our training size down to around 2,000 samples and we discussed the limitations of this in Section 4.

Future work may be able to address this by utilising approaches to impute or estimate values for samples that have missing features but we believe there is a large opportunity here to build a much more complete and contemporary dataset to allow researchers to test whether more complete 'simple' metadata can achieve similar performances to either traditional signal processing genre prediction models (Oramas et al., 2018) (Kostrzewa et al., 2021), or to pre-processed metadata features such as the Echonest features that we evaluated in this work.

In addition, the FMA is a very useful dataset as it represents copyright free music so that researchers are free to directly use the music samples in their work. However, a downside of this is that the music that is represented is often not contemporary and perhaps not fully representative of the kinds of music people consume. This can clearly be seen by the fact that $\tilde{2}0\%$ of the original dataset is made up of the 'Experimental' genre and only $\tilde{5}\%$ is made of 'Pop' music. This, coupled with the problem of every artist only being associated with a single genre (a problem that exists in every other dataset we looked at too) leads to our conclusion that there is desperate need for better, more up-to-date datasets to properly explore the potential of simple metadata features as genre predictors.

While theoretically existing datasets such as the Million Songs Dataset (Bertin-Mahieux et al., 2011) could be combined with others utilising services such as AcousticBrainz ID mapping for multiple different services[10], a lot of these datasets will continue to fall short of the existing pitfalls we have discussed and will lack rich descriptive metadata. We envision a new dataset that could include information such as date of recording, age of band, location of recording, song writer, producer, and many more, and hope more research can be done into creating this.

5.1.2 Better Ensemble Approaches

We had hoped to be able to use ensemble approaches to combine some of our weaker classifiers to achieve better results. Whilst improving on existing datasets might also help with this, we think that in its current state there is still a lot of potential for applying better ensemble approaches even to the existing flawed datasets.

With more time, we would have liked to more thoroughly investigate the shortcomings of our existing models on predicting specific genres and look at approaches to build multiple weak classifiers that could perform binary classification of, for example, whether a track was Rock, or not Rock. Using an approach such as Adaboost, we could then look to combine these models together and be able to re-weight misclassified tracks with more confidence. Being able to understand the strength of each of our classifiers for each specific genre could also allow us to build a stronger voting classifier that weighted each model's prediction of genre according to its perceived strength for classifying that genre.

We would also have liked to investigate how evening out the distribution of genres in our training set could impact the classification of genres. It is highly likely that the lesser represented genres were often misclassified as they had less training examples. The tradeoff of building classifiers that can more accurately predict these niche genres however, is that these genres are, generally, less prevalent in the wild too.

There are further interesting avenues of research to investigate approaches to be able to incorporate features such as artist name which suffer from one-to-one mappings into the per-track genre classification problem. This may be improved by more accurate datasets that don't rely on tagging all songs by an artist with the same genre, but we think there could also be potential approaches such as incorporating the results we saw in Table 3 into any ensemble approaches with an appropriate weight, and perhaps introducing some random noise to repeated artist names to reduce the risk of overfitting.

5.1.3 Moving Away from Genre Classification

Finally, recent commercial offerings such as YouTube Music[11] have begun to move away from using genre as a useful marker especially for providers wishing to understand their consumers listening behaviour. In fact, even as early as 2006, some research was questioning the utility of genre and genre classification as an area of research in general(McKay and Fujinaga, 2006).

One interesting avenue for future work could be to utilise unsupervised machine learning techniques such as clustering to try and form similar groups of music without relying on the, at times, reductive labelling of genre. This could then be evaluated and analysed by more qualitative assessment with both musical experts and non-expert consumers to understand which features were helping to contribute to more coherent groupings of music, as well as better recommendations for users. In addition, this could be tied to and compliment more recent work that looks to improve the explainability of music recommendation systems. (Afchar et al., 2022; Afchar, 2023)

6 Supplemental Materials

The Github repository hosting our code can be found here. In order to run any of the code you will need to download the FMA dataset and insert the files 'features.csv', 'tracks.csv' and 'raw_tracks.csv' into the fma_metadata direc-

[10] https://labs.acousticbrainz.org/million-song-dataset-echonest-archive/

[11] https://blog.youtube/news-and-events/explore-your-2023-recap-on-youtube-music/

tory.[12]. The rest of our results can be found in our google sheet here. Sheet 1 contains the preliminary results from our single feature models. Sheet 2 contains the results of our single and combined feature models given different subsets of the original FMA dataset. The 'Boosted Models' sheet contains the results from our boosted models and Sheet 3 contains some additional results from our combined models. Sheet 5 contains the best accuracy results from all of our single, combined, boosted, and voting models as shown in Table 10.

References

Darius Afchar. 2023. *Interpretable Music Recommender Systems*. Ph.D. thesis, Sorbonne Université.

Darius Afchar, Alessandro Melchiorre, Markus Schedl, Romain Hennequin, Elena Epure, and Manuel Moussallam. 2022. Explainability in music recommender systems. *AI Magazine*, 43(2):190–208.

Ritesh Ajoodha, Richard Klein, and Benjamin Rosman. 2015. Single-labelled music genre classification using content-based features. In *2015 pattern recognition association of south africa and robotics and mechatronics international conference (prasa-robmech)*, pages 66–71. IEEE.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. The music streaming sessions dataset. In *The World Wide Web Conference*, WWW '19, page 2594–2600, New York, NY, USA. Association for Computing Machinery.

Débora C. Corrêa and Francisco Ap. Rodrigues. 2016. A survey on symbolic data-based music genre classification. *Expert Systems with Applications*, 60:190–210.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. 2016. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*.

Alex Hern. 2014. Spotify acquires music data firm the echo nest. Accessed [2024-05-07].

De Rosal Ignatius Moses Setiadi, Dewangga Satriya Rahardwika, Eko Hari Rachmawanto, Christy Atika Sari, Candra Irawan, Desi Purwanti Kusumaningrum, Nuri, and Swapaka Listya Trusthi. 2020. Comparison of svm, knn, and nb classifier for genre music classification based on metadata. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 12–16.

Daniel Kostrzewa, Piotr Kaminski, and Robert Brzeski. 2021. Music genre classification: Looking for the perfect network. In *Computational Science – ICCS 2021*, pages 55–67, Cham. Springer International Publishing.

Edith Law and Luis Von Ahn. 2009. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1197–1206.

Tao Li, Mitsunori Ogihara, and Qi Li. 2003. A comparative study on content-based music genre classification. pages 282–289.

Cory McKay and Ichiro Fujinaga. 2006. Musical genre classification: Is it worth pursuing and how can it be improved? In *ISMIR*, pages 101–106.

Sergio Oramas, Francesco Barbieri, Oriol Nieto, and Xavier Serra. 2018. Multimodal Deep Learning for Music Genre Classification. In *Transactions of the International Society for Music Information Retrieval*.

Sergio Oramas, Luis Espinosa-Anke, Aonghus Lawlor, Xavier Serra, and Horacio Saggion. 2016a. Exploring Customer Reviews for Music Genre Classification and Evolutionary Studies. In *17th International Society for Music Information Retrieval Conference*.

Sergio Oramas, Luis Espinosa-Anke, Aonghus Lawlor, et al. 2016b. Exploring customer reviews for music genre classification and evolutionary studies.

Dewangga Satriya Rahardwika, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Christy Atika Sari, Ajib Susanto, Ibnu Utomo Wahyu Mulyono, Erna Zuni Astuti, and Amiq Fahmi. 2020. Effect of feature selection on the accuracy of music genre classification using svm classifier. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 7–11.

George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302.

[12] This is due to the file size limit on GitHub repos