Working Title: Light it up: Predicting Song Features to Motivate Lighting Choice

Ahmed Abdalla, Mark Bechthold, Tristan Saucedo

May 7, 2024

1 Abstract

2 Introduction

A staple of any college dorm room is the RGB LED lighting strip. These lights often come with the supposed ability to sync their hue and intensity to the ambient soundscape. But, as any college student can tell you, this preprogrammed function is often too inaccurate and awkward to be of any use. We applied machine learning methods to help solve this problem.

We first trained an algorithm to predict the quantitative audial characteristics (pitch and timbre) at any point in a song using what the algorithm had previously heard. The recursive nature of this problem naturally led us to explore recurrent neural networks (RNN). We benchmarked this approach against a fully connected neural networks (FCNN) and logistic regression (LR) to confirm that a RNN is uniquely suited to tackling this problem. We also predict qualitative classifications (genre) using K-Means classification on a wider set of features.

Armed with this predictive ability, we built a pipeline to translate our model outputs into RGB color sweeps on an LED strip light controlled by a Raspberry Pi. We combined Western notions of tonality with our own subjectivity to systematically extract relative notions of tension and movement in a piece. For a given genre, these values map to specific locations on pre-determined color bars. Crucially the lighting design anticipates upcoming musical changes before they occur. The model is subsequently adjusted using the new audio data allowing us to accomplish accurate song feature prediction and solve a perennial problem for college students everywhere.

3 Related Works

Music Information Retrieval (MIR) is a field of research that focuses on the development of algorithms and methods for automatically extracting, analyzing, and representing musical data. Some of the most common tasks in MIR include audio segmentation, audio classification, music generation, music recommendation, music similarity, and music structure analysis. We focused on music generation (pitch and timbre prediction, leveraging time-stamped data), and explored genre prediction.

3.1 Genre Prediction

Previous research in MIR has included the use of machine learning techniques such as Support Vector Machines (SVMs) and Neural Networks (NNs) to classify different types of audio data [11, 9]. However, they are limited in their ability to recognize complex musical structures, like harmony and melody, and when used for genre prediction face difficulty distinguishing between similar genres.

Modified versions of the K-Means algorithm have been used to understand and predict musical genres [12]. In addition to audio features such as pitch and timbre, authors have also used lyrical data and sound intensity to achieve higher fidelity classification [3]. Such work used augmented versions of The Million Song Dataset, the same dataset we use in our training.

We noted that the most accurate genre prediction uses random forest algorithms, but elected to implement a different classification pipeline, specifically a more diverse dataset [10, 1]. The best models currently use comparatively small dataset, with less songs from more distinct genres to prevent confusion in the algorithm.

3.2 Music characteristic prediction

Deep learning techniques have also been used to address the task of MIR. Convolutional Neural Networks (CNNs) have been used to recognize musical patterns from spectrograms [2], while RNNs have been used to generate new music from a dataset of existing musical pieces [6]. Researchers have also used deep neural networks to augment digital musical instruments, such as the electric keyboard [8]. Additional work leverages RNNs to generate music in different styles or genres, or user input to modify the generated music [4]. These generative models are similar to our goal, where we are given the music that has been played and predict the upcoming pitch and timbre features of music.

The best generative models use RNN's, and [5]. These models tend to achieve high performance, but haven't been leveraged for light shows.

4 Dataset and Features

We trained our models using subsets of the Million Song Dataset. We were drawn to this dataset because it contained features that we hypothesized could be naturally translated into aesthetic lighting. Other datasets we considered were composed of pre-selected song samples designed to simplify machine learning methods. We specifically focused on two $N \times 12$ arrays which contained pitch and timbre information. Here N refers to the number of segments in a song, where songs are broken up into segments using an unknown algorithm that identifies melodic changes. These segments have a mean time interval of 0.33 and a standard deviation of 0.23. The 12 columns refer to the 12 pitch classes in Western equal temperament. For a given octave, playing one note from each pitch class yields a chromatic scale. As such we refer to the pitch matrix, where values are normalized such that the greatest entry per row is 1, as the chroma matrix.

The timbre matrix contains Mel-Frequency Cepstral Coefficients (MFCCs). The human ear is an imperfect sound sensor making raw sonic intensity a poor metric for perceived loudness. The Mel scale corrects for this by adjusting intensity for human perception. The Fourier transform power spectrum is mapped to the log-Mel scale for each of the twelve chromatic pitches. The inverse Fourier transform of these values gives the MFCCs in our dataset. These type of cepstral features are common in speech recognition and instrument classification research, often referred to as quefrency alanysis [7, 13].

For our FCNN and RNN we ignored metadata and only used these time-series features. For K-Means genre prediction, we included metadata (song length, tempo, time signature, and key), but removed smaller genres with overlap to make the problem more tractable, keeping only hip-hop, punk, dance and electronica, jazz and blues, soul and reggae, classic pop and rock, folk, classical, pop, and metal. We chose these 10 genres specifically to build on previous work, using well documented methodology. Additionally, we normalized the genre size and explored smaller subsets of genres. Some challenges we faced were the subjective nature of genres ("pop" vs "classic pop and rock" vs "rock and indie" is especially challenging). This data was also averaged over the entire length of the song, and thus not discretized over time.

5 Methods

There were two distinct parts of our analysis. First, we used K-means and Principle Component Analysis to explore our data, and better understand differences in genres of music. Then we used two neural networks to predict pitch and timbre features of songs in real time.

K-Means is a clustering algorithm that can be used to group data points into clusters based on their similarity. This can be useful for music genre prediction because it can identify patterns in the data that are indicative of certain genres. For example, by using k-means, a model could be created that identifies patterns in the frequency, intensity, and timbre of different songs to identify what genre they belong to. Using metadata of songs, such as length, tempo, and time signature, allows us to pull apart different genres.

Principal Component Analysis (PCA) is a dimensional reduction technique that can be used to reduce the number of variables in a dataset, or to find the most explanatory vector multidimensional data. This can be useful for music genre prediction because it can help to identify the most important features in the data that are indicative of different genres. We used PCA to explore the variance of genre subsets. To do this, we used a subset of the million song dataset described above.

A **Classical Neural Network** was used to predict the upcoming pitch, and timbre, in our time sequence data. To do this we created a sliding window of 20 segments as our input example and tried to predict the next segment in the song. Effectively our input was a 480 dimensional vector of inputs comprised of the the 20 sets of 12 chroma and 12 timbre values flattened together. We then predicted the output which was a 24 dimensional vector corresponding to the 12 chroma and 12 timbre values of the next segment. We picked a sliding window of 20 because it encoded 5 seconds a given song and seemed a relevant amount of data for predicting the next value in the sequence. Using this data we then split it into a training, dev, and test set comprising 70%, 20%, and 10%. With this architecture we started by establishing a baseline where we performed linear regression without any intermediate layers in our network. Then we created a neural net by adding intermediate layers to the regression of varying size using ReLU between layers. Using the python package Ray tune, we sampled various hyperparameter configurations for the learning rate, batch size, and the number of intermediate layers and their associated sizes. We sampled learning rates from a uniform log from 0.1 to 0.0001 and our batch size to be 32, 64, or 128. We ran 5 such samples for 100 epochs, and to assess our performance we used Mean Square Error, according to the following equation:

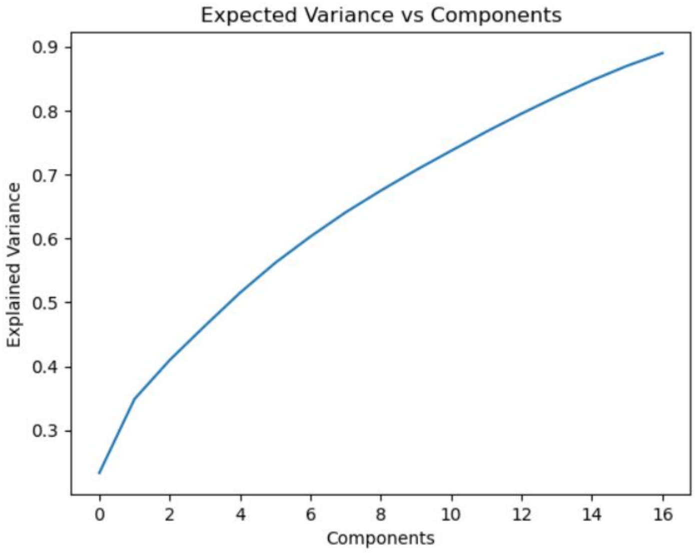
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2$$

Additionally, we used a **Recurrent Neural Network**, leveraging the sinusoidal nature of music to make better predictions. Using the same 12 features for timbre and 12 features for pitches to represent a single time segment, we padded our inputs to the maximum length of a song and for a song with n time segments we passed in the first n-1 times segments into the recurrent neural net to predict time steps 2 through n. We randomly shuffled the data, and divided our data set into a training set consisting of 70% of the data, a dev set consisting of 20% of the data and a test set consisting of 10% of the data set. Due to time considerations we were forced to run our model on about a fourth of these inputs after dividing into these categories. We further tuned our model using the python package Ray tune, randomly choosing our hidden layer to have dimensions between 2^2 and 2^9, choosing between 1 and 6 layers for the stacked RNN, sampling a uniform log between 0.1 and 0.0001 for our learning rate, and randomly choosing between 32, 64, and 128 examples in a given batch. We only ran three such samples due to time considerations. We ran the samples for 50 epochs, and used Mean Squared Error to assess our performance.

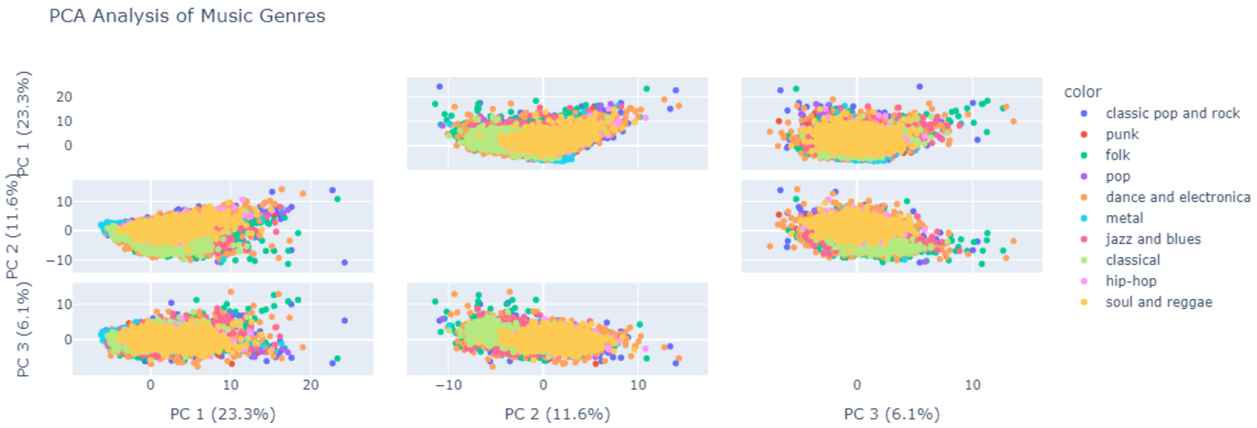
6 Experiment/Results/Discussion

First, we will explore our data using PCA to better understand some of the difficulties in our analyses. We can see from figure 1 that the expected variance is captured poorly by the first few principle components. This implies a weak relationship between the variables, and doesn't allow PCA to adequately reduce the dimensions of the data without significant loss of information.

To predict genre, we utilized k-means. While not the only thing that dictates the final light show output, genre has a notable impact on the tone of music. Using our PCA from figure 1, we found that some genres were more distinguishable than others, and implementing k-means on subsets of the data resulted in accurate predictions (figure 2 (a)). However, on the full data set of genres, k-means struggled to achieve such high accuracy. Since we normalized the sizes of all genres, randomly guessing would be expected to have an accuracy of 10%. Thus, our model does 3 times better than random. Given the similarity of the features, explored in figure 1, this result is compelling.

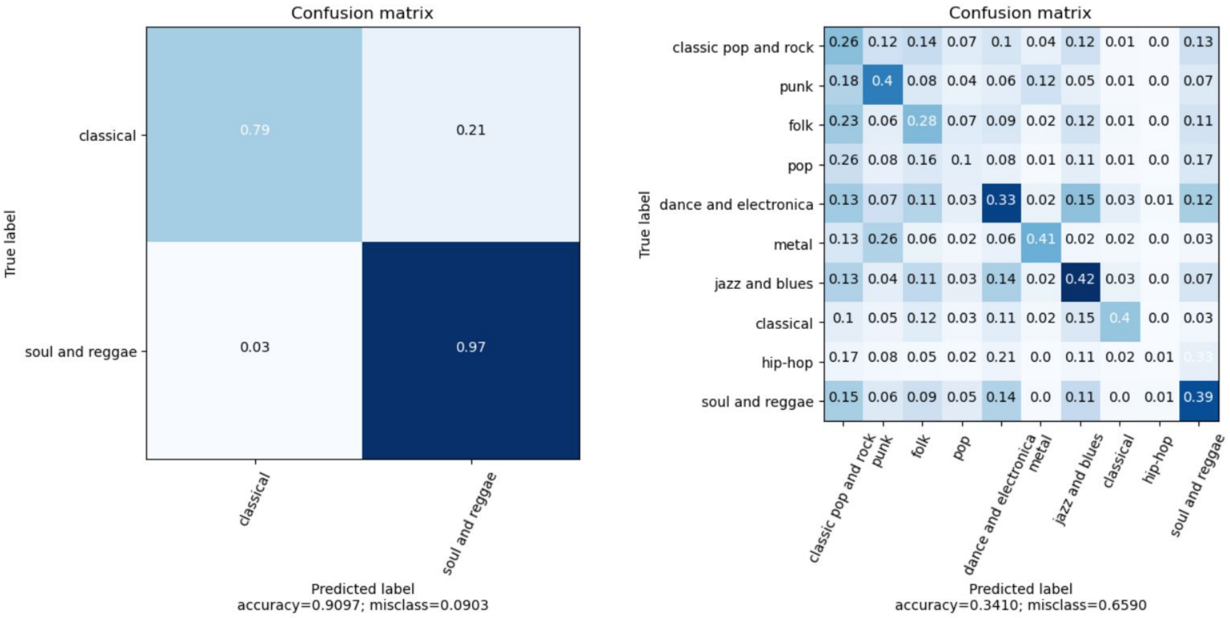


(a)



(b)

Figure 1: (a) Explained variance is low, and the relationship between principle components is nearly linear. Thus, PCA has difficulty reducing the dimensions of this dataset. (b) Visualizing the PCA, we can see that there is significant overlap between genres, even when graphed against their principle components with the largest variance.

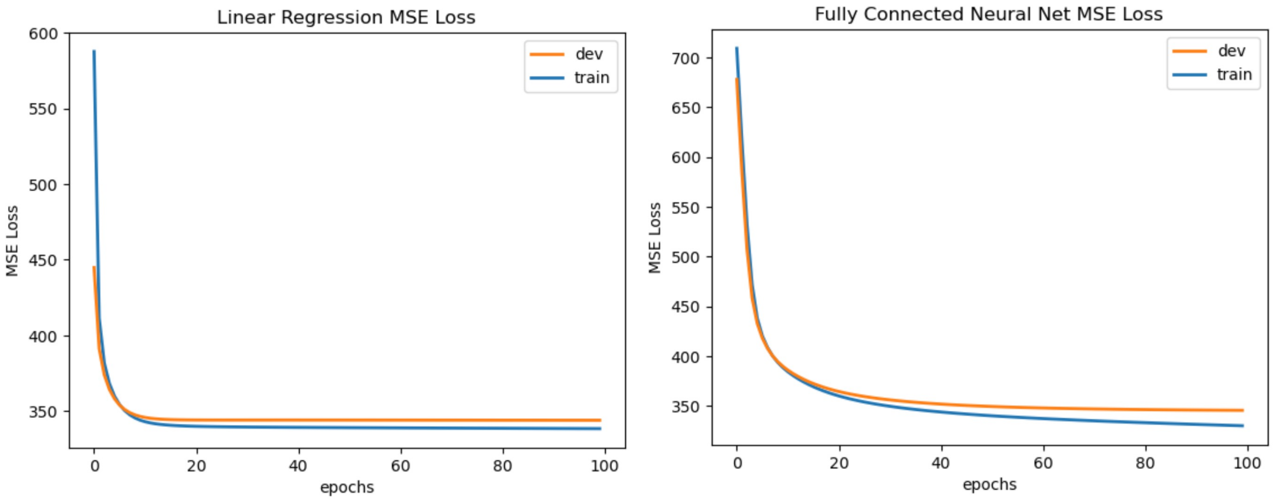


	classical	soul and reggae
classical	0.79	0.21
soul and reggae	0.03	0.97

	classic pop and rock	punk	folk	pop	dance and electronica	metal	jazz and blues	classical	hip-hop	soul and reggae
classic pop and rock	0.26	0.12	0.14	0.07	0.1	0.04	0.12	0.01	0.0	0.13
punk	0.18	0.4	0.08	0.04	0.06	0.12	0.05	0.01	0.0	0.07
folk	0.23	0.06	0.28	0.07	0.09	0.02	0.12	0.01	0.0	0.11
pop	0.26	0.08	0.16	0.1	0.08	0.01	0.11	0.01	0.0	0.17
dance and electronica	0.13	0.07	0.11	0.03	0.33	0.02	0.15	0.03	0.01	0.12
metal	0.13	0.26	0.06	0.02	0.06	0.41	0.02	0.02	0.0	0.03
jazz and blues	0.13	0.04	0.11	0.03	0.14	0.02	0.42	0.03	0.0	0.07
classical	0.1	0.05	0.12	0.03	0.11	0.02	0.15	0.4	0.0	0.03
hip-hop	0.17	0.08	0.05	0.02	0.21	0.0	0.11	0.02	0.01	0.33
soul and reggae	0.15	0.06	0.09	0.05	0.14	0.0	0.11	0.0	0.01	0.39

(a) (b)

Figure 2: Confusion matrices for k means with different genres. (a) Using two distinct genres shows the ability of k-means to split the feature space between genres. The genres ”classical” and ”soul and reggae” were chosen based off of PCA values which highlighted the difference in features between the two genres. (b) Predictions for all genres proved more challenging, where the diagonal is correct predictions for each genre. Our model is accurate, with the highest probabilities in nearly all classes.



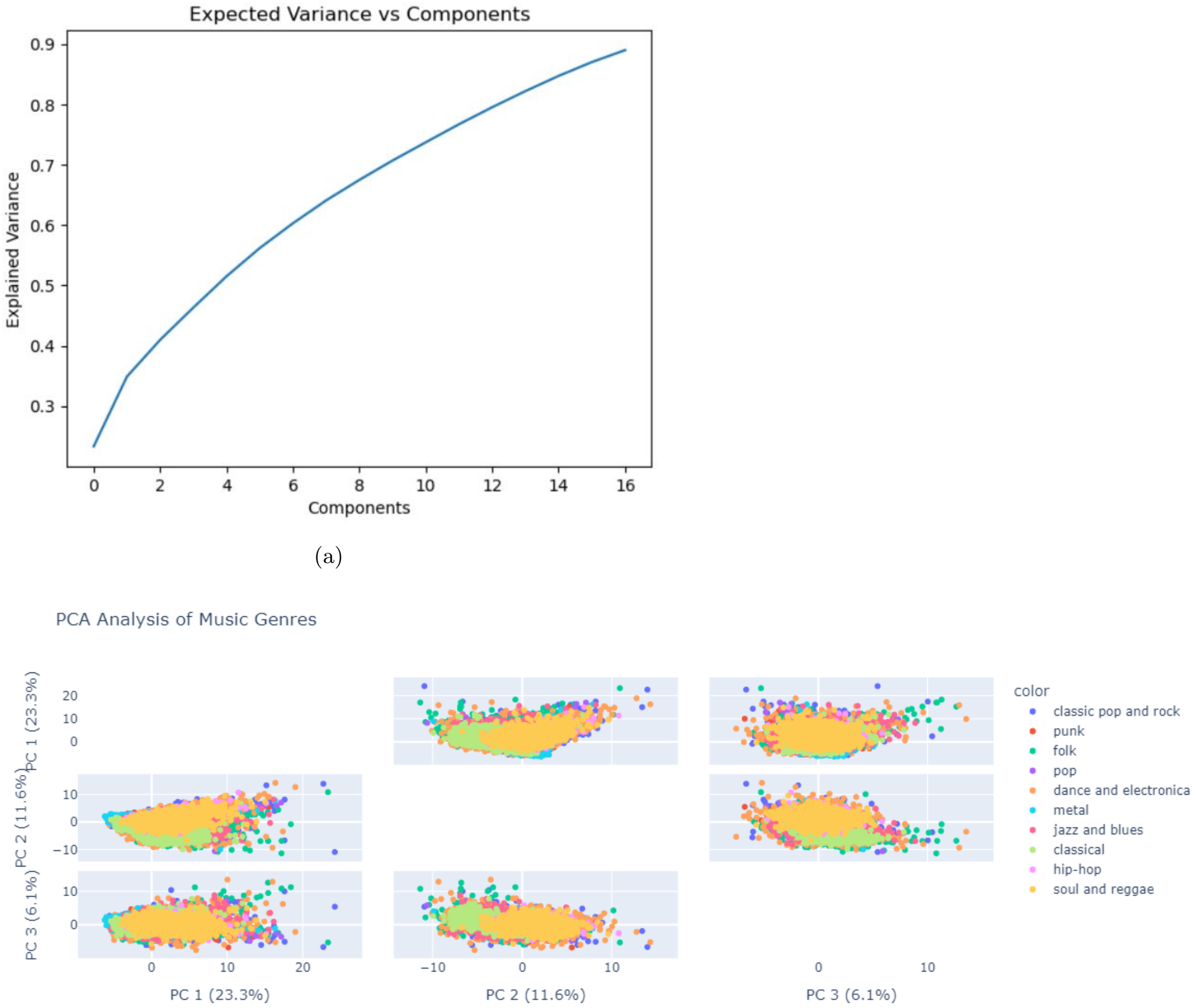
(a) (b)

Figure 3: Linear Regression and Fully Connected Neural Net. (a) For Linear Regression, we

To confirm our findings, we randomly initialized multiple starting values to ensure that our models were actually learning, instead of converging to random values. In conjunction with MSE loss, which is inherently tied to the magnitude of vectors, we can see that both linear regression and our fully connected neural net preformed admirably. While the linear regression model converges faster than the fully connected neural net, both preformed significantly better than random initialization with a loss of 1450. This indicates that the data can be modeled using basic machine learning methods, but large absolute loss values leaves room for improvement.

(a) (b)

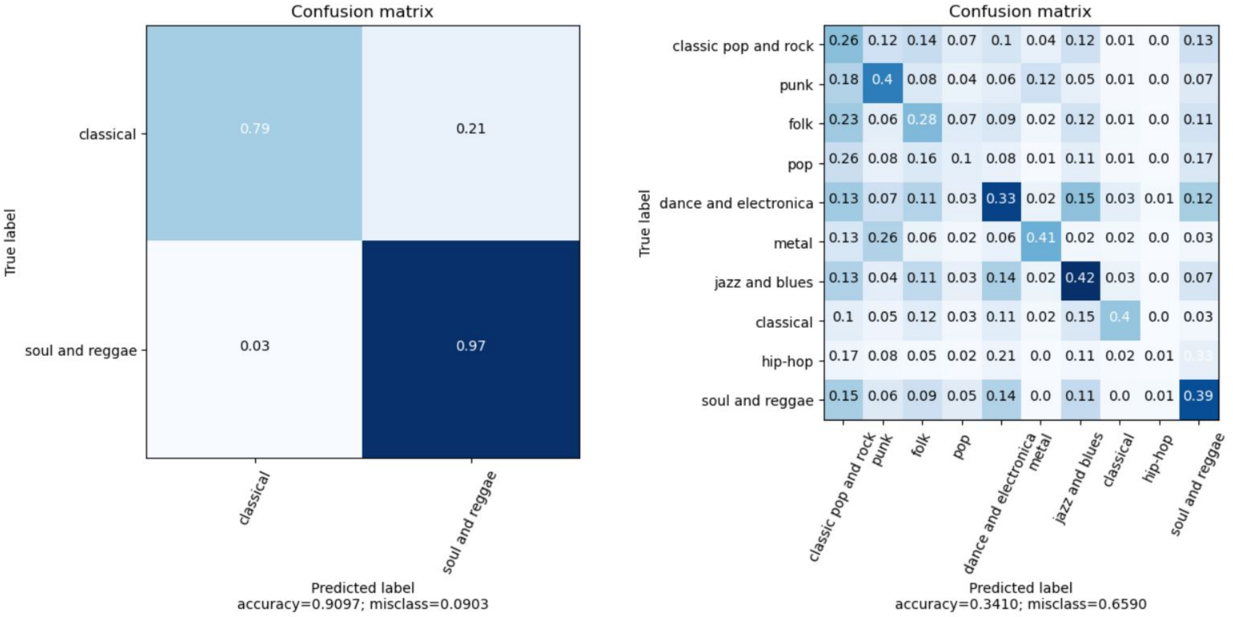
Figure 4



(a)

(b)

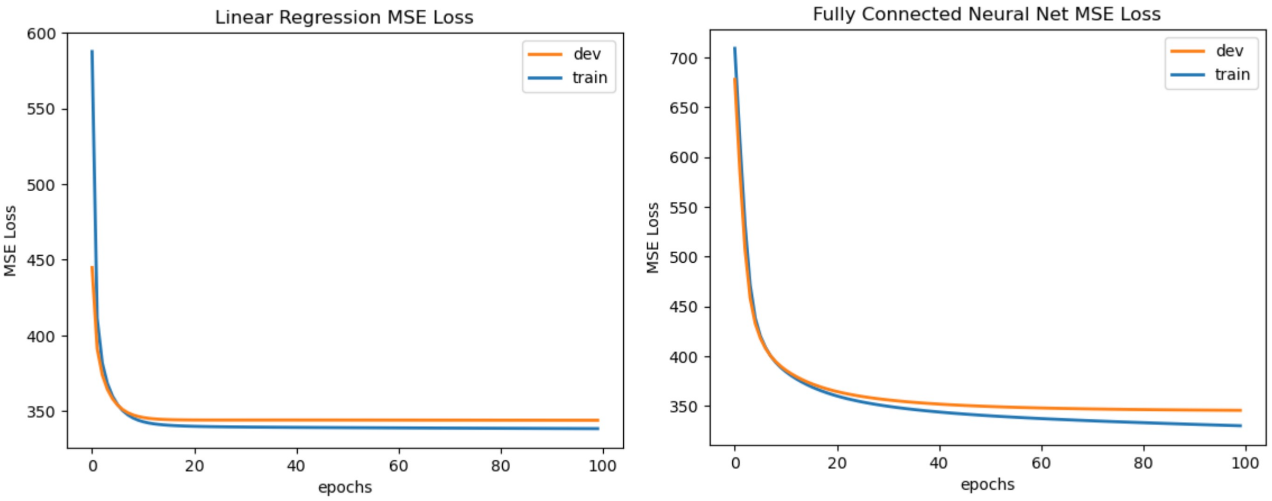
Figure 5: (a) Explained variance is low, and the relationship between principle components is nearly linear. Thus, PCA has difficulty reducing the dimensions of this dataset. (b) Visualizing the PCA, we can see that there is significant overlap between genres, even when graphed against their principle components with the largest variance.



(a)

(b)

Figure 6: Confusion matrices for k means with different genres. (a) Using two distinct genres shows the ability of k-means to split the feature space between genres. The genres ”classical” and ”soul and reggae” were chosen based off of PCA values which highlighted the difference in features between the two genres. (b) Predictions for all genres proved more challenging, where the diagonal is correct predictions for each genre. Our model is accurate, with the highest probabilities in nearly all classes.



(a)

(b)

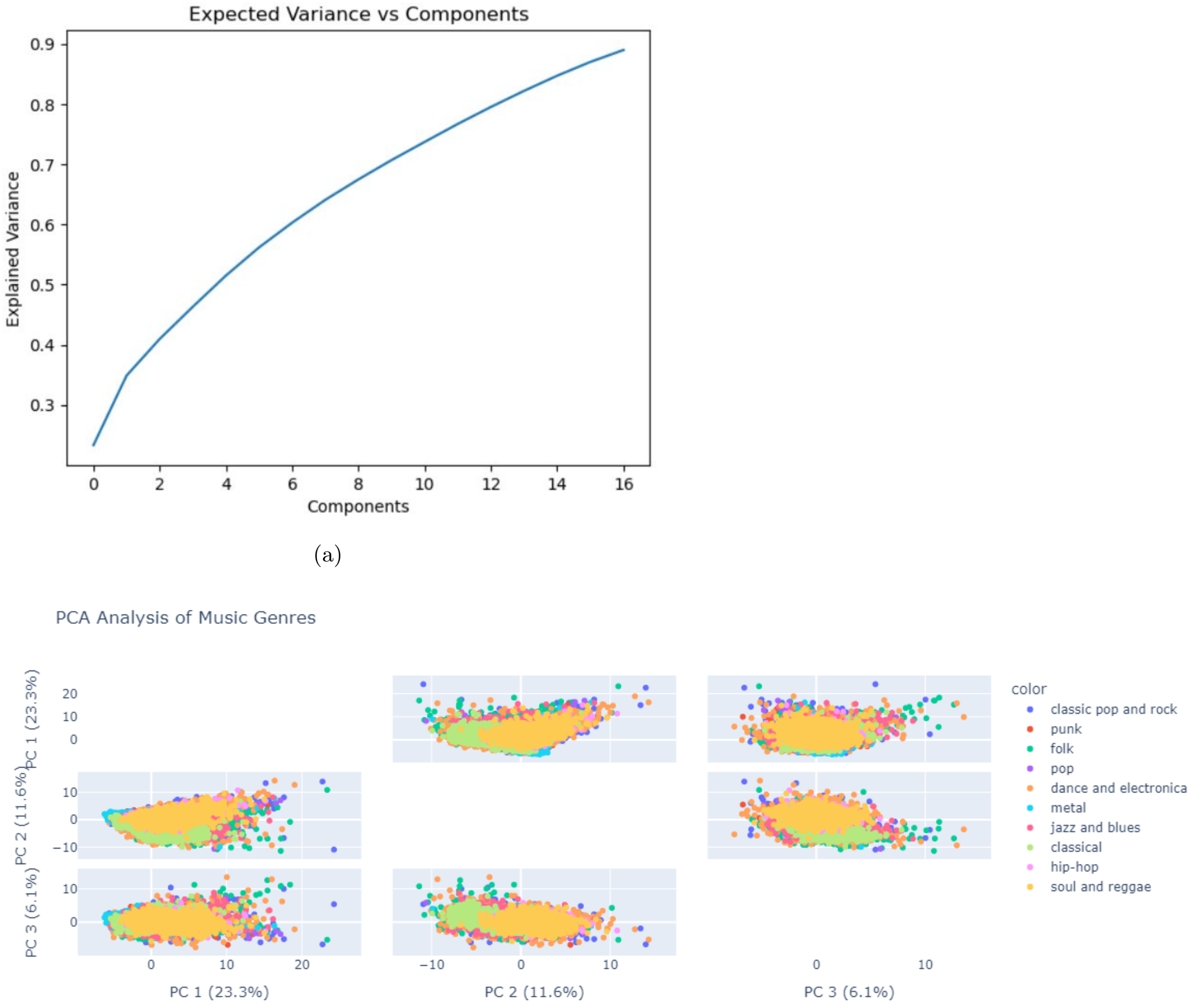
Figure 7: Linear Regression and Fully Connected Neural Net. (a) For Linear Regression, we

To confirm our findings, we randomly initialized multiple starting values to ensure that our models were actually learning, instead of converging to random values. In conjunction with MSE loss, which is inherently tied to the magnitude of vectors, we can see that both linear regression and our fully connected neural net preformed admirably. While the linear regression model converges faster than the fully connected neural net, both preformed significantly better than random initialization with a loss of 1450.

(a)

(b)

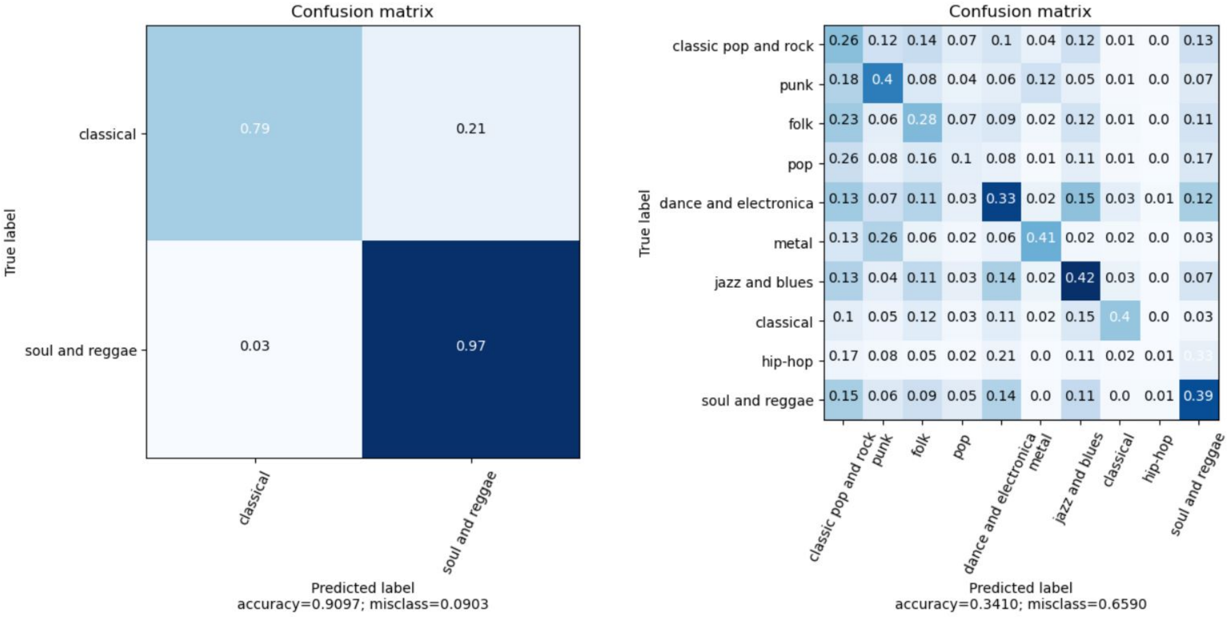
Figure 8



(a)

(b)

Figure 9: (a) Explained variance is low, and the relationship between principle components is nearly linear. Thus, PCA has difficulty reducing the dimensions of this dataset. (b) Visualizing the PCA, we can see that there is significant overlap between genres, even when graphed against their principle components with the largest variance.

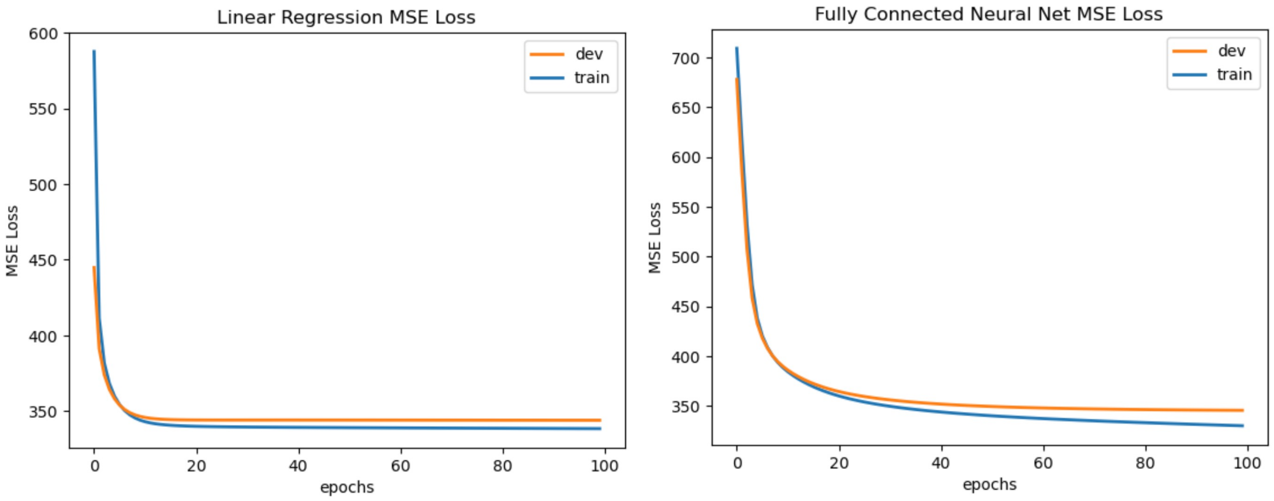


	classical	soul and reggae
classical	0.79	0.21
soul and reggae	0.03	0.97

	classic pop and rock	punk	folk	pop	dance and electronica	metal	jazz and blues	classical	hip-hop	soul and reggae
classic pop and rock	0.26	0.12	0.14	0.07	0.1	0.04	0.12	0.01	0.0	0.13
punk	0.18	0.4	0.08	0.04	0.06	0.12	0.05	0.01	0.0	0.07
folk	0.23	0.06	0.28	0.07	0.09	0.02	0.12	0.01	0.0	0.11
pop	0.26	0.08	0.16	0.1	0.08	0.01	0.11	0.01	0.0	0.17
dance and electronica	0.13	0.07	0.11	0.03	0.33	0.02	0.15	0.03	0.01	0.12
metal	0.13	0.26	0.06	0.02	0.06	0.41	0.02	0.02	0.0	0.03
jazz and blues	0.13	0.04	0.11	0.03	0.14	0.02	0.42	0.03	0.0	0.07
classical	0.1	0.05	0.12	0.03	0.11	0.02	0.15	0.4	0.0	0.03
hip-hop	0.17	0.08	0.05	0.02	0.21	0.0	0.11	0.02	0.01	0.33
soul and reggae	0.15	0.06	0.09	0.05	0.14	0.0	0.11	0.0	0.01	0.39

(a) (b)

Figure 10: Confusion matrices for k means with different genres. (a) Using two distinct genres shows the ability of k-means to split the feature space between genres. The genres ”classical” and ”soul and reggae” were chosen based off of PCA values which highlighted the difference in features between the two genres. (b) Predictions for all genres proved more challenging, where the diagonal is correct predictions for each genre. Our model is accurate, with the highest probabilities in nearly all classes.



(a) (b)

Figure 11: Linear Regression and Fully Connected Neural Net. (a) For Linear Regression, we

To confirm our findings, we randomly initialized multiple starting values to ensure that our models were actually learning, instead of converging to random values. In conjunction with MSE loss, which is inherently tied to the magnitude of vectors, we can see that both linear regression and our fully connected neural net preformed admirably. While the linear regression model converges faster than the fully connected neural net, both preformed significantly better than random initialization with a loss of 1450.

(a) (b)

Figure 12

7 Conclusion/Future Work

Predicting the upcoming features of music is a challenging problem, even for most humans with musical training. We found that RNN's were the best for this specific predictive test. The

8 Contributions

Our names are listed beside the components of the assignment that we contributed to. In general, we all worked together, checked each other's code, and supported each other in debugging, writing, and model understanding.

-
-

References

[1] Chaudhury, M., Karami, A., and Ghazanfar, M. A. Large-scale music genre analysis and classification using machine learning with apache spark. *Electronics 11*, 16 (2022), 2567.

[2] Costa, Y. M., Oliveira, L. S., and Silla Jr, C. N. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing 52* (2017), 28–38.

[3] Dawen Liang, H. G., and O'Connor, B. Music genre classification with the million song dataset, 2011. Last accessed 9 December 2022.

[4] Hadjeres, G., and Nielsen, F. Interactive music generation with positional constraints using anticipation-rnns, 2017.

[5] Hui Tang, Y. Z., and Zhang, Q. The use of deep learning-based intelligent music signal identification and generation technology in national music teaching. *Frontiers in Psychology* (2022).

[6] Lu, W. T., Su, L., et al. Transferring the style of homophonic music using recurrent neural networks and autoregressive model. In *ISMIR* (2018), pp. 740–746.

[7] Maliki, I., et al. Musical instrument recognition using mel-frequency cepstral coefficients and learning vector quantization. In *IOP Conference Series: Materials Science and Engineering* (2018), vol. 407, IOP Publishing, p. 012118.

[8] Martin, C. P., Ellefsen, K. O., and Torresen, J. Deep predictive models in interactive music, 2018.

[9] Pelchat, N., and Gelowitz, C. M. Neural network music genre classification. *Canadian Journal of Electrical and Computer Engineering 43*, 3 (2020), 170–173.

[10] Tang, H., Zhang, Y., and Zhang, Q. The use of deep learning-based intelligent music signal identification and generation technology in national music teaching. *Frontiers in psychology 13* (2022), 762402.

[11] Thiruvengatanadhan, R. Music genre classification using svm, 2018.

[12] Tzanetakis, G., and Cook, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing 10*, 5 (2002), 293–302.

[13] Ubaidi, U., and Dewi, N. Voice pattern recognition using mel-frequency cepstral coefficient and hidden markov model for bahasa madura. In *Journal of Physics: Conference Series* (2019), vol. 1375, IOP Publishing, p. 012057.