Final Report

CS 4641

View on GitHub Return Home

Introduction and Background

Whether it be calm music for studying, energetic music while exercising, or just music for everyday activities, everyone enjoys listening to music. The use of machine learning to accurately predict the song genre is commonly used by music platforms such as Spotify, Apple Music, YouTube Music, etc. Various algorithms using reinforcement learning such as training a Deep Q Network [1], Baysian Networks to evaluate both song novelty and audio content [2], and Neural Networks in combination with filtering [3] have all been explored.

Problem Definition and Motivation

Being able to understand the genre of music that one listens to provides great insight, and enhances their auditory experience. This allows them to develop a sense of connection within their favorite genres, and encourages discovery and exploration outside of their musical taste. There have been attempts to identify musical genre by using pure audio content-based methods, but results have a perceived lower quality than algorithms that are prone to biases[4]. Uncommon genres are more likely to be given an inaccurate genre tag, resulting in them being less likely to be discovered by listeners.

Problem statement: Music listeners need an accurate and unbiased music genre classification system to encourage connection and discovery among others.

Methods and Preprocessing

Our project utilizes the below Kaggle dataset. It includes over 125 different genres of songs, and each song has audio features such as “valence”, “tempo”, “liveliness”, “speechiness”, etc. associated with it that has been retrieved from Spotify’s API.

- dataset link

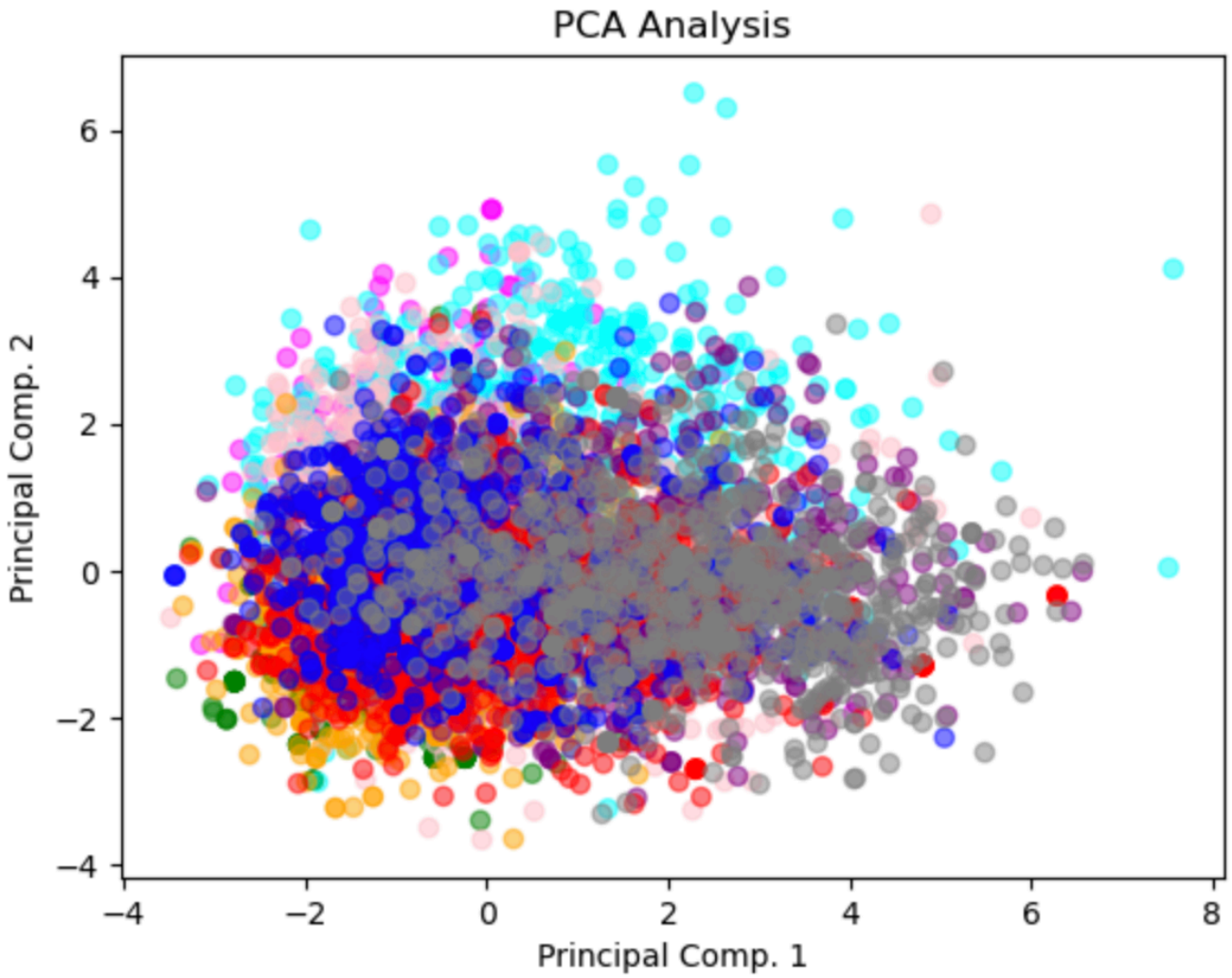
Pre Processing: PCA

The Kaggle dataset has over 100,000 songs, but we only need 3000 samples for the project. To get the samples, we randomly sampled 3000 songs from the dataset. From then, we used `StandardScaler` to standardize and scale the features. Using scikit-learn, we used PCA to perform Principal Component Analysis to reduce the dimensionality to both 2 and 3 components.

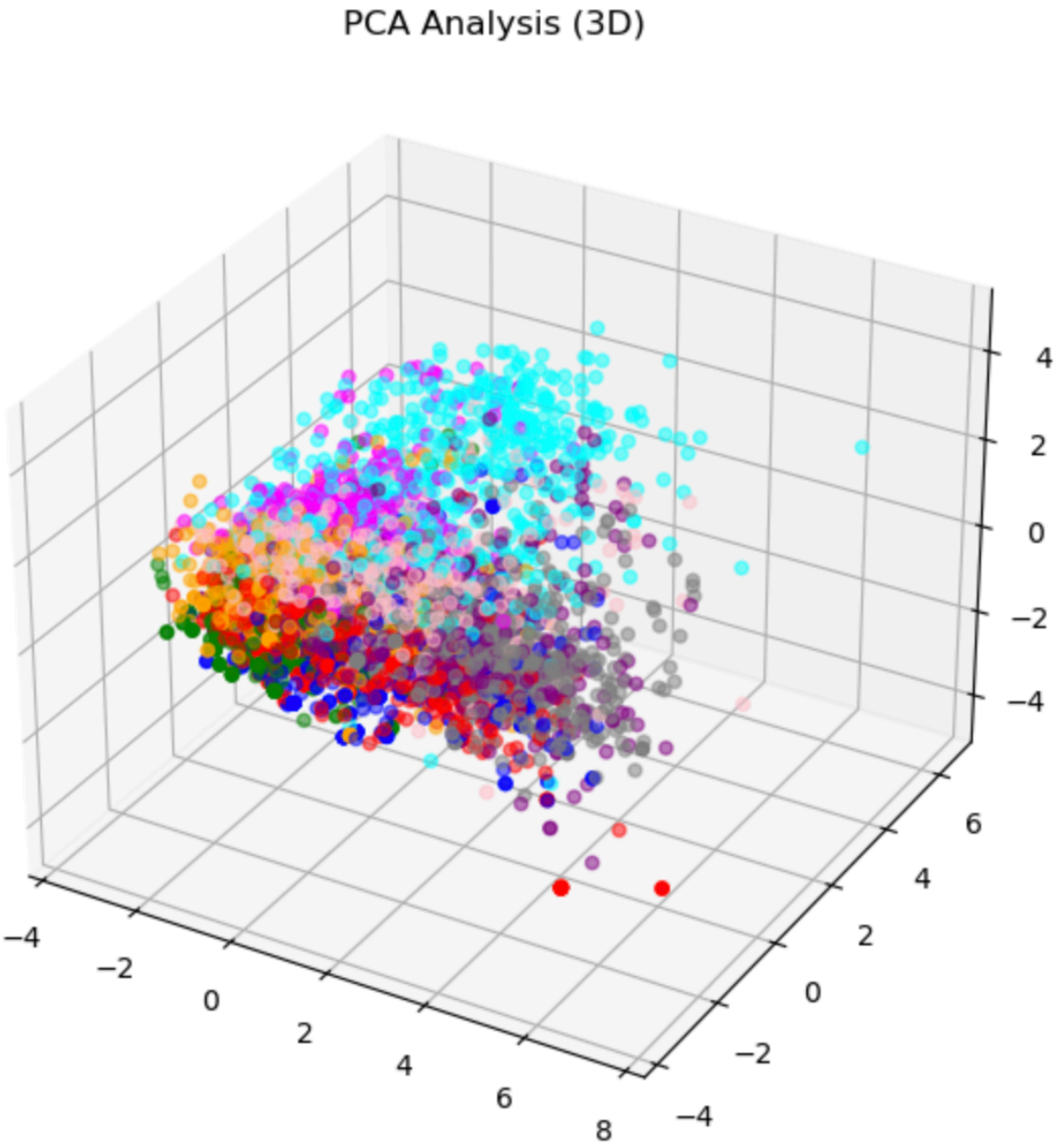
Dimensionality

Since varied songs can produce varied feature data, we are seeking to improve interpretability by applying dimensionality reduction techniques, ultimately maintaining consistency in the extracted data. PCA (Principal Component Analysis), available in scikit-learn as `PCA`, is effective for this, especially to visualize data in two or three dimensions.

- PCA with 2 components:



- PCA with 3 components



The above plot is colored by genre.

Method 1: Supervised Learning: Random Forest

In our midterm progress report on the Spotify genre classification machine learning model, we have chosen to transition from a decision tree algorithm to a random forest classifier. This decision was underpinned by the need to handle the expansive and complex dataset effectively, which comprises 10,000 songs categorized into 114 distinct genres. The random forest model is advantageous in this context due to its inherent capability to manage high-dimensional data and its robustness against overfitting, which is common in decision tree models. This ensemble method

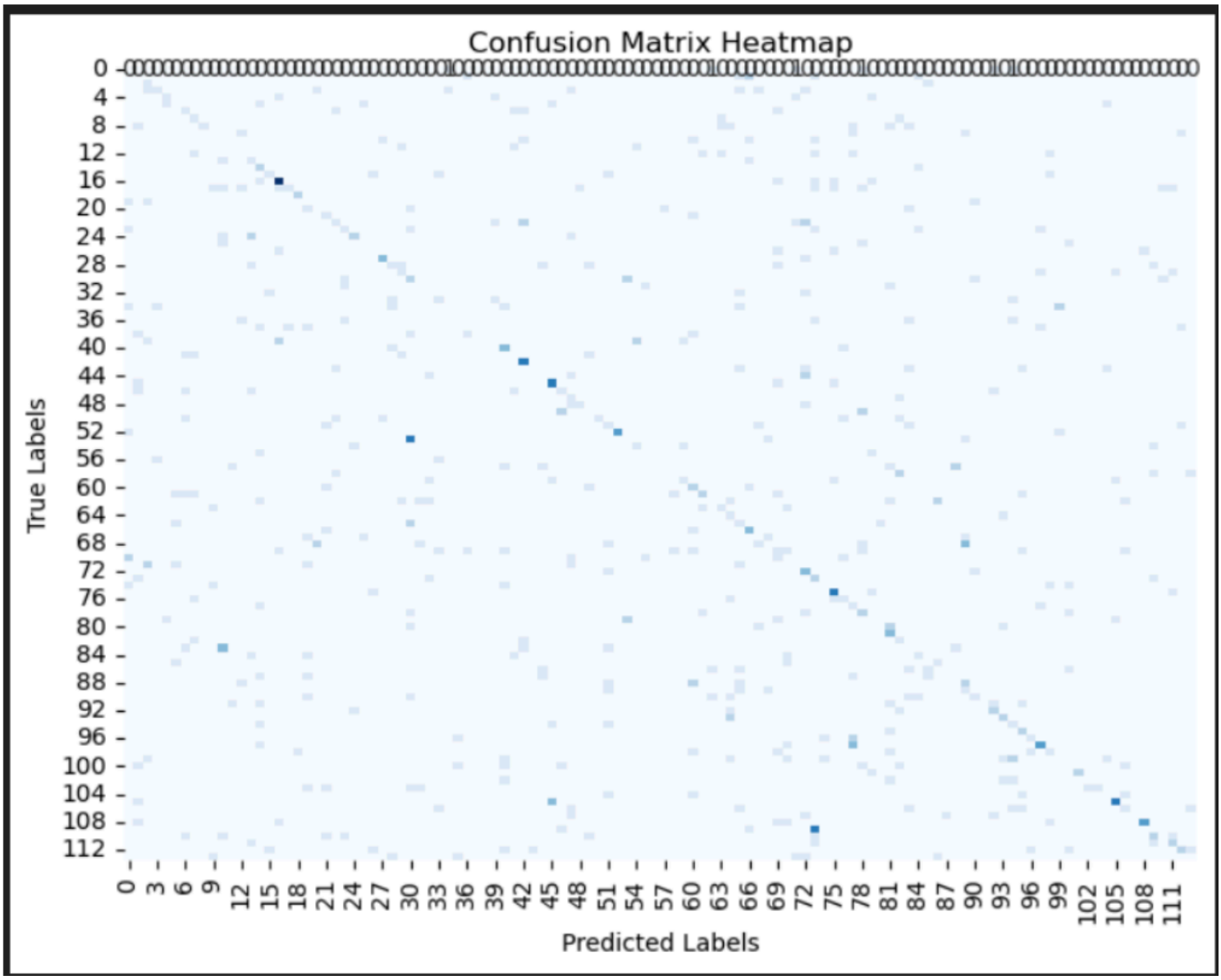
combines multiple decision trees to improve classification accuracy by reducing variance and bias, leading to a more generalized model.

Results Part 1

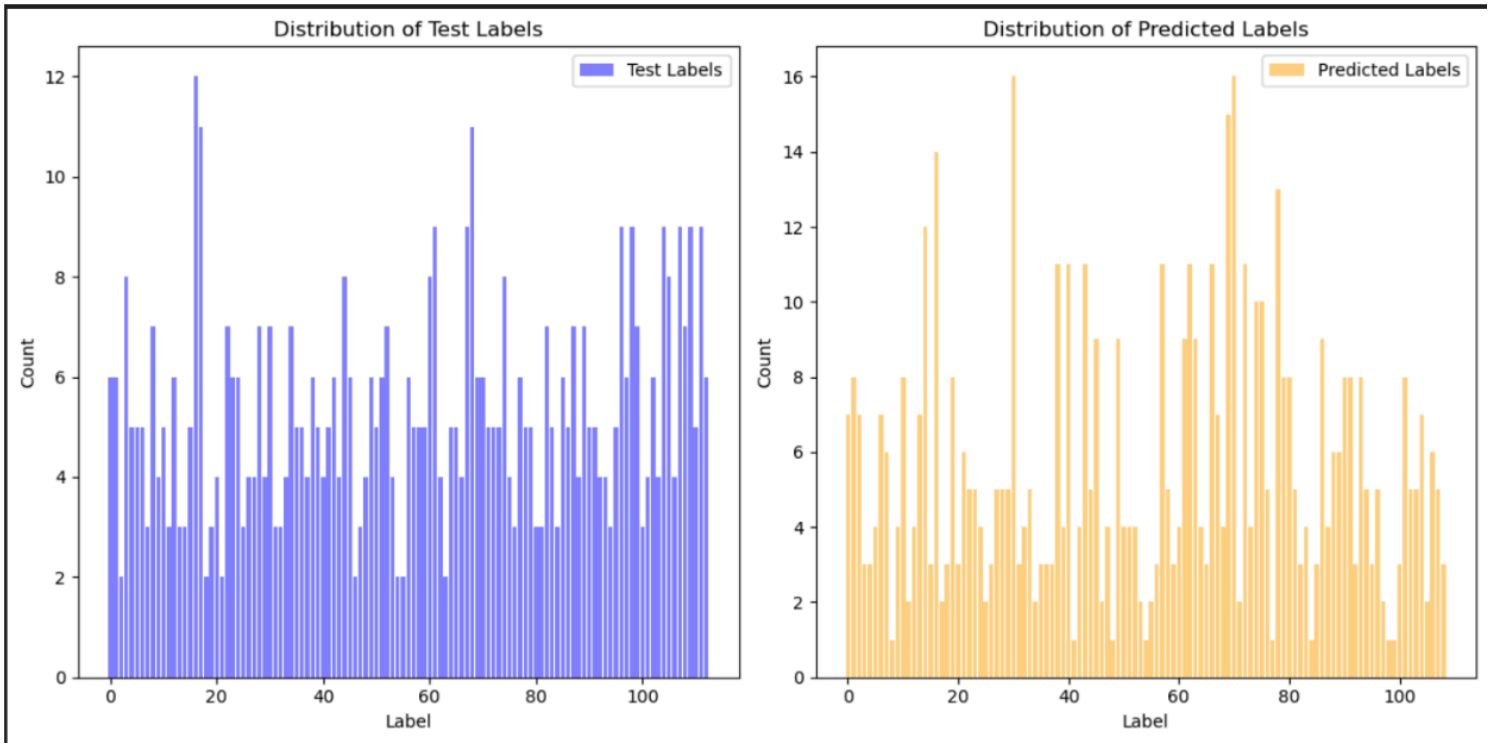
Upon evaluation, the random forest classifier yielded an accuracy score of 0.20 with weighted averages of 0.21 for precision, 0.20 for recall, and 0.18 for F1-Score. The macro averages are 0.19 for precision, 0.21 for recall, and 0.17 for F1-Score. These metrics, although revealing the model's present limitations in accurately classifying such a wide array of genres, provide a critical insight into the underlying complexity of the genre classification task. The relatively low scores indicate that the model struggled to fit the condensed features, obtained post-Principal Component Analysis, into independent, non-overlapping categories.

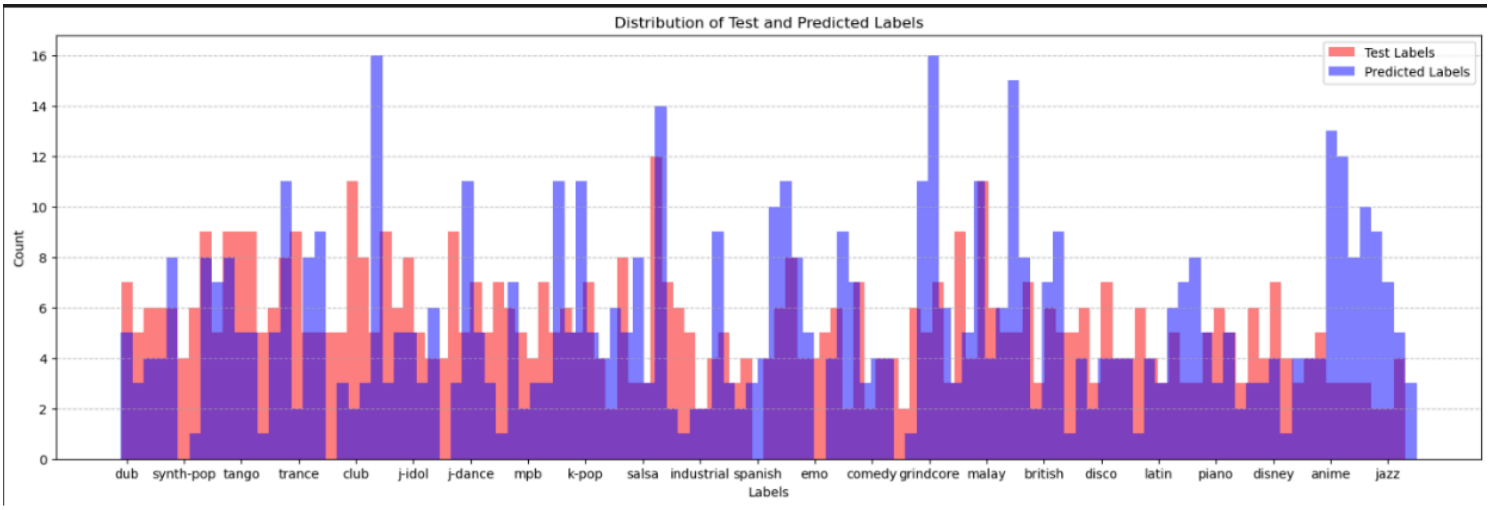
Visualization

The visualizations of the model's performance shown below reflect these classification challenges. They highlight the significant overlap among genre categories and the model's tendency to misclassify songs, especially in genres with subtle distinction features. The algorithm shows some understanding of music genre classification but its accuracy is low, which can be attributed to potential bias towards certain genres and an imbalance in the training data.



- The heatmap shows that the algorithm has a general idea of how to classify as we can see the diagonal, which represents correct predictions, but it has low accuracy.





- This distribution shows that the algorithm has bias to certain genres which can be attributed to imbalance in the training data.

Discussion

Given these outcomes, our model demonstrates a need for further refinement and optimization. The poor performance can be attributed to multiple factors such as the high dimensional space after applying PCA, which might have led to the loss of important genre-distinguishing features, as well as an imbalance in the training data leading to potential bias towards certain genres. To address this, the next steps will involve exploring alternative feature selection methods that preserve more genre-specific information and balancing the training data. Additionally, experimenting with other machine learning models or deep learning approaches may provide better handling of the complex genre feature interactions. To enhance the classification accuracy,

we plan to revisit the feature extraction and selection phases, and trial different classification algorithms. These actions are aimed at developing a more accurate and reliable genre classification model for Spotify's diverse music dataset.

Improvement Since Midterm Report

We have improved our Random Forest Classifier since the midterm by reducing the classification scope and properly incorporating Principal Component Analysis (PCA). We reduced the dataset and used the Random Forest to only classify 9 genres instead of the previous 114 that we were doing. This helped the accuracy because the 3000 points in our previous dataset was not nearly enough to allow our model to understand how to classify each of the 114 different genres. In the current version, we have only 9 genres that are being classified and the Random Forest utilizes the principle components given by the PCA to help it with its classification. We utilized grid search to help find the most optimal hyperparameters which gave us the hyperparameters of Bootstrap = True, Max Depth = 30, Max Features = 'sqrt', Min Sample Leaves = 1, and Min Sample Split =5. Evaluation of the Random Forest Classifier yields an accuracy score of 0.41 with weighted averages of 0.40 for precision, 0.41 for recall, and 0.40 for F1-Score. The macro averages yield the same results with 0.40 for precision, 0.41 for recall, and 0.40 for F1-Score. Although we reduced the scope of the classification and utilized PCA to help with the classification, the algorithm still does not yield a high accuracy. This could be due to the size of the database being small or not enough variability within each genre causing the algorithm to make more mistakes in classification.

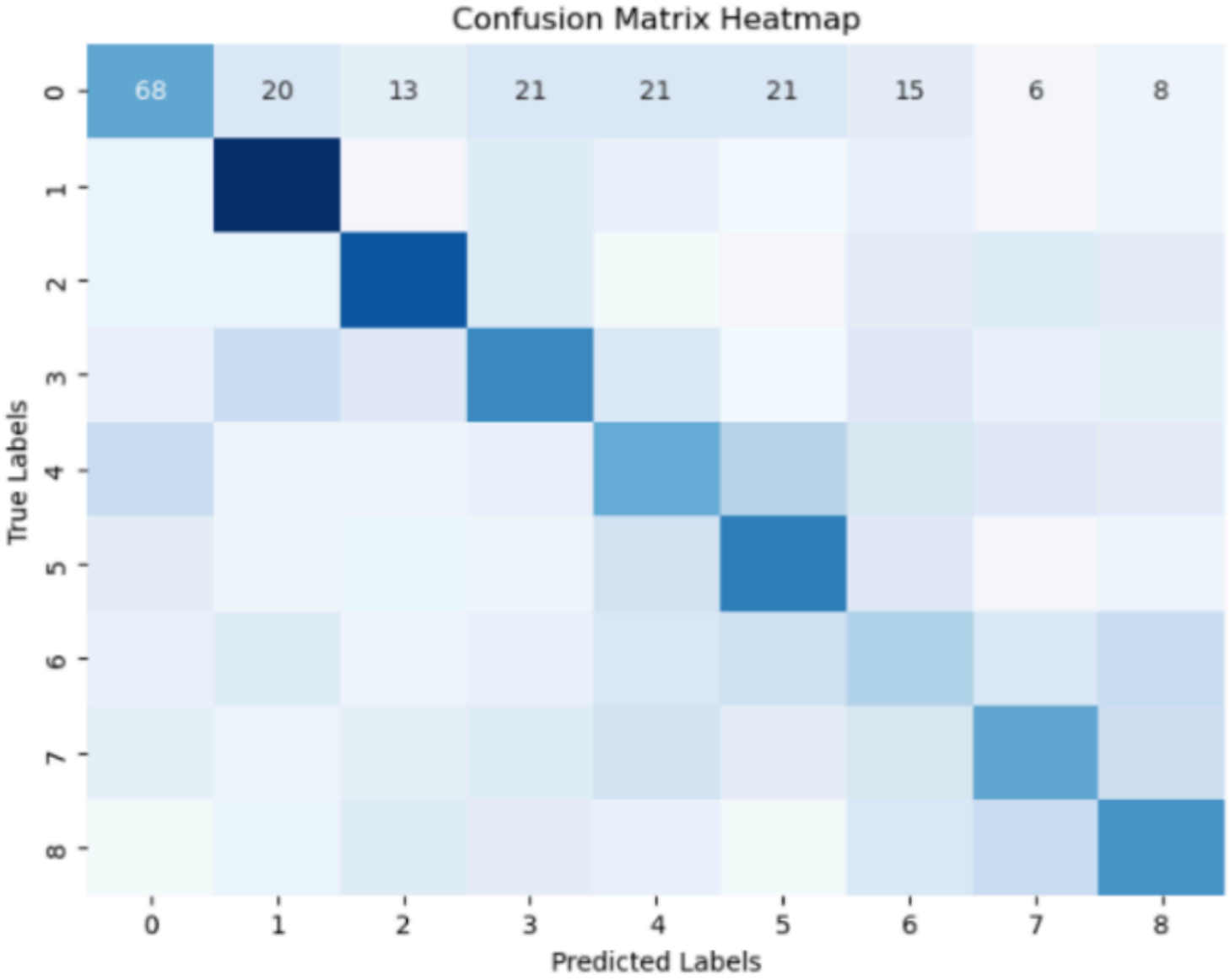
Target Genres:

```
['pop', 'dance', 'rock','post-teen-pop', 'hip-hop', 'indie-pop', 'songwriter', 'emo'
```

Visualization (post midterm)

Once we ran the same algorithm as previously, we were able to attain an accuracy of 0.41, with weighted averages of 0.40 precision, 0.41 recall, and 0.40 for F1-score as seen below:

	precision	recall	f1-score	support
1	0.40	0.36	0.38	193
2	0.52	0.63	0.57	197
3	0.53	0.53	0.53	201
5	0.41	0.38	0.39	210
6	0.31	0.29	0.30	218
7	0.44	0.48	0.46	189
8	0.23	0.22	0.23	190
9	0.35	0.32	0.34	210
10	0.41	0.45	0.43	192
accuracy			0.41	1800
macro avg	0.40	0.41	0.40	1800
weighted avg	0.40	0.41	0.40	1800



Heatmap for Random Forest *after* reducing the total number of genres in the dataset

Method 2: Unsupervised Learning: Gaussian Mixture Model

We decided to use a Gaussian Mixture model due to its accommodation to mixed membership of points in multiple clusters. This is particularly useful for music, where a song might blend elements of multiple genres. It accounts for the covariance structure of data and the centers of lantent Gaussians. We used scikit-learn's `GaussianMixture` to implement this algorithm. GMM is beneficial because it provides a probabilistic cluster assignment, which is more nuanced than hard clustering. From our literature review, we are seeing that GMM has been proven to demonstrate a more flexible clustering approach, accommodating songs that exhibit characteristics of multiple genres, reflecting the reality of music genre blending.

Results

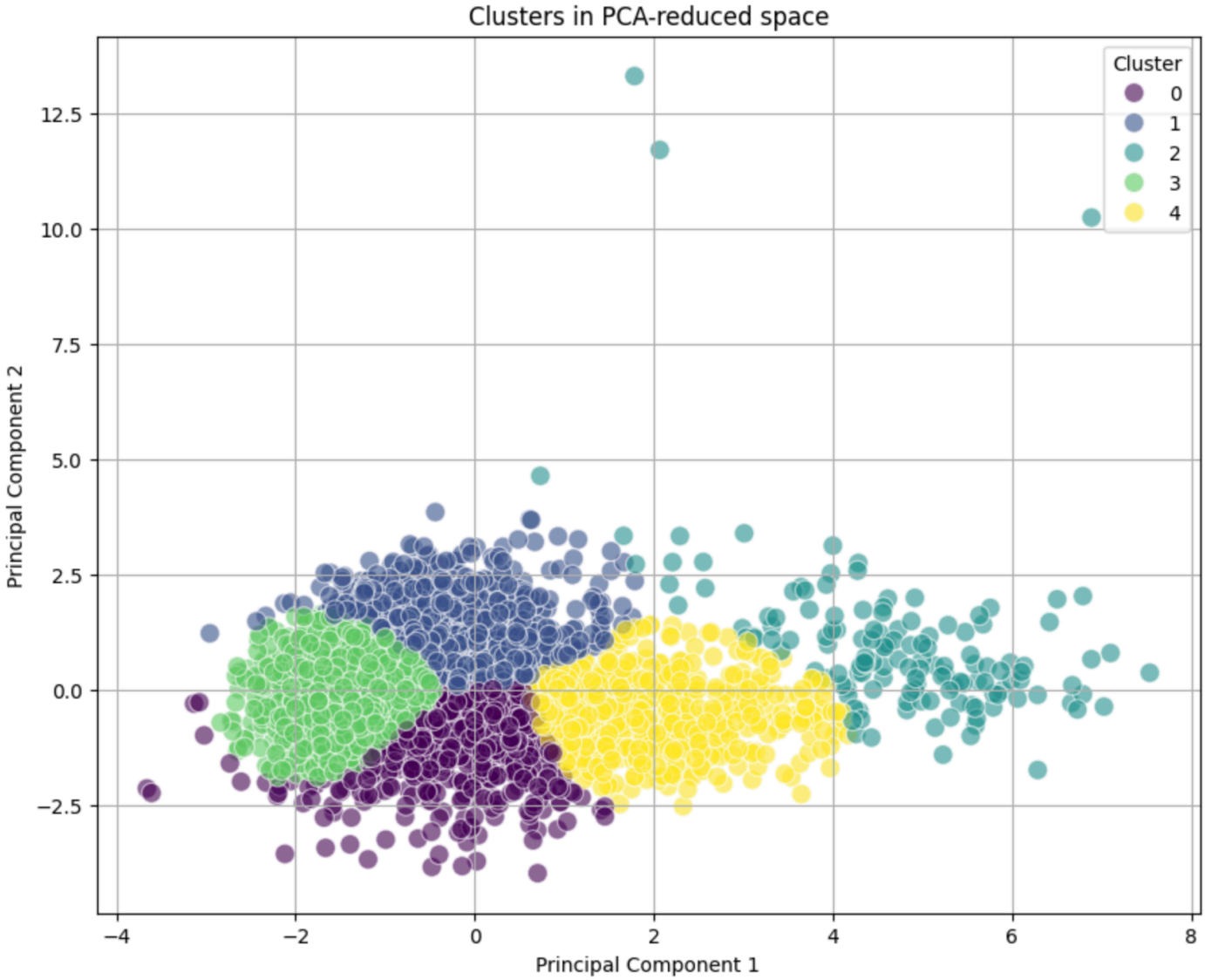
For the evaluation, we got a Silhouette score of 0.33 which indicates very fair clustering. We also checked our algorithm to see how far it deviates from the actual ground truth of the musical genres by trying to cluster sub genres into larger genre sets. Evaluating this gave us clustering metrics of 0.04 Spearman's rank correlation, 0 Adjusted Rand Index, and 0.07 Normalized Mutual Information.

```
other            1735
edm               571
rock              285
children           63
reggae             54
indie-pop          51
malay              36
world-music        35
afrobeat           35
j-idol             34
k-pop              34
classical          34
german             33
Name: simplified_genre, dtype: int64
[0 1 2 3 4]
Spearman's rank correlation: 0.04
Adjusted Rand Index: -0.00
Normalized Mutual Information: 0.07
```

Discussion

This tells us that while the model was able to form clusters based on feature similarity, there is minimal correlation with the actual music genre. This is shown by the low scores in the quantitative metrics. This means that the clustering might not effectively capture genre distinctions, potentially due to the diverse nature of the genres and their overlap in musical features. Another source of the low quantitative metrics is the dataset, which is highly unbalanced in the favor of the genre label 'other', leaving different genres with much lower representation. Moving forward, a way to improve this algorithm would be to have a larger and more balanced dataset where each genre has adequate representation which would allow for more accurate genre distinctions.

Visualization



Method 3: Unsupervised Learning: Neural Nets

Our second algorithm used to implement our Spotify music genre classification is Neural Nets. In this algorithm, we follow the same preprocessing steps as in the Random Forest as we reduce the scope of our classification to only 10 genres. We also use PCA to get the principal components and then pass them along to a GMM which gives us our labels. For the implementation of this algorithm, we utilized the Python library sklearn and used the MLPRegressor. For the Neural Net model itself, we did a lot of experimentation for the hyperparameters and landed on a learning rate of 0.001, a hidden layer size of (14, 5), and ReLU as our activation function.

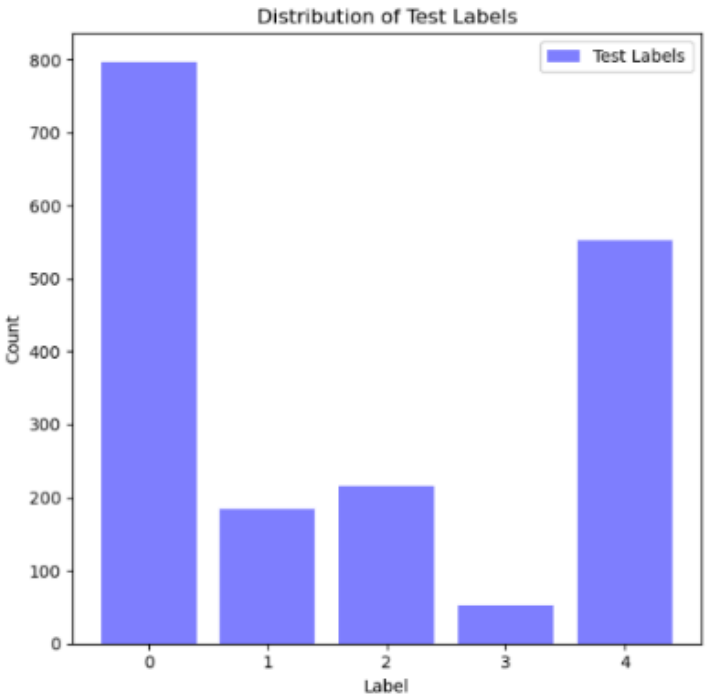
Results and Discussion

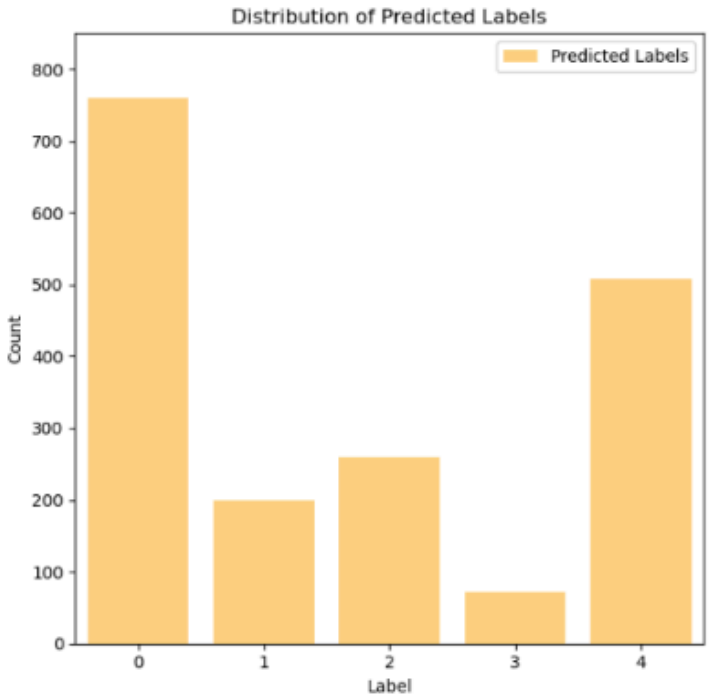
Evaluation of the Neural Net algorithm yields an accuracy score of 0.90 with weighted averages of 0.91 for precision, 0.90 for recall, and 0.90 for F1-Score. The macro averages yield the same results with 0.63 for precision, 0.66 for recall, and 0.64 for F1-Score. Overall, we are pretty satisfied with the

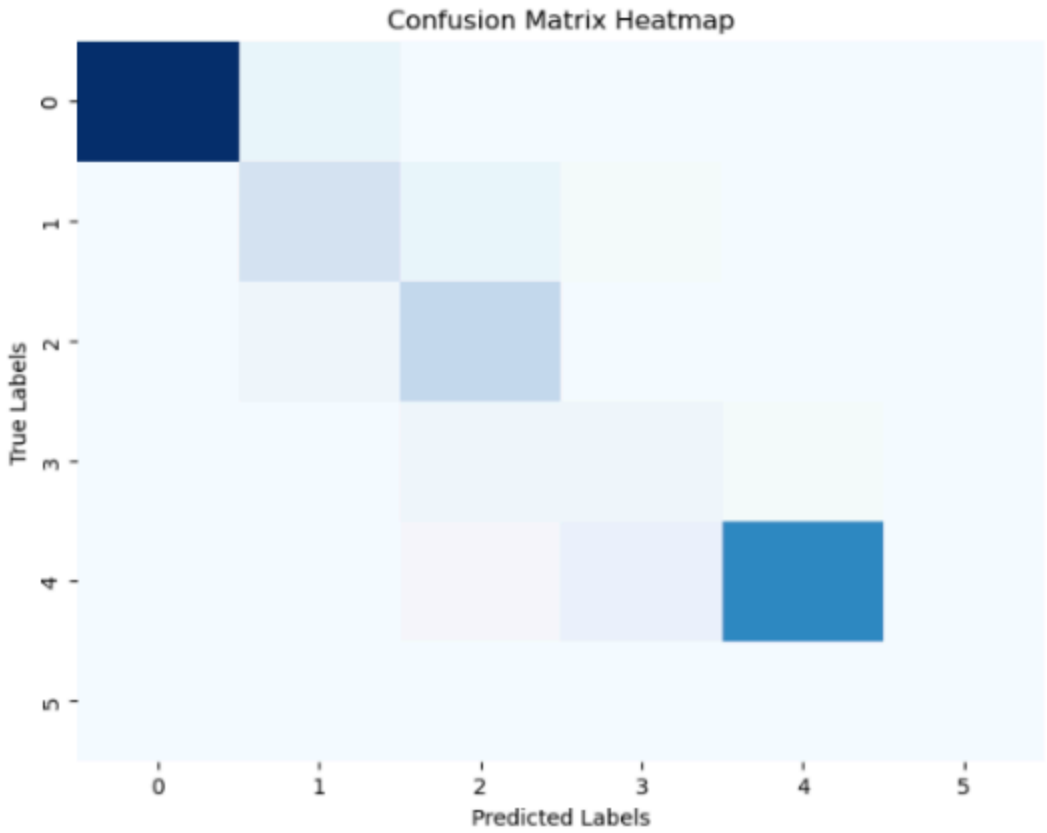
performance of our Neural Net algorithm as it resulted in a high accuracy. Some factors that contributed to this high accuracy are PCA which allowed for effective dimension reduction, hyperparameter experimentation leading to optimal hyperparameters, effective activation function and neural network architecture, and an effective GMM.

	precision	recall	f1-score	support
0.0	1.00	0.95	0.98	796
1.0	0.70	0.77	0.73	184
2.0	0.73	0.88	0.80	216
3.0	0.33	0.46	0.39	52
4.0	0.99	0.91	0.95	552
5.0	0.00	0.00	0.00	0
accuracy			0.90	1800
macro avg	0.63	0.66	0.64	1800
weighted avg	0.91	0.90	0.90	1800

Visualization







Comparison

When comparing all the algorithms, it is clear that the algorithm with the best performance was the Neural Net with an accuracy of 90%. The GMM algorithm was used to cluster the sub genres into larger genre sets which was used by the Neural Net and contributed to its success. This shows that

the GMM implementation was also quite effective. The Random Forest Classifier had poorer results in comparison with an accuracy of 40%.

Next Steps

For Next Steps, one thing that we would definitely want to use is a larger and more balanced dataset with adequate representation in each model which would increase our algorithms' performances. We could also further broaden the scope of our inputs by trying to look for other features that would help in capturing genre distinctions. By doing so, we hope to contribute to the continued enjoyment of music.

References

- IEEE format, all references must have an intext citation

[1] F. Tomasi et al., "Automatic Music Playlist Generation via simulation-based reinforcement learning," Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Aug. 2023. doi:10.1145/3580305.3599777

[2] X. Wang, Y. Wang, D. Hsu, and Y. Wang, "Exploration in interactive personalized music recommendation," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 11, no. 1, pp. 1–22, Aug. 2014. doi:10.1145/2623372

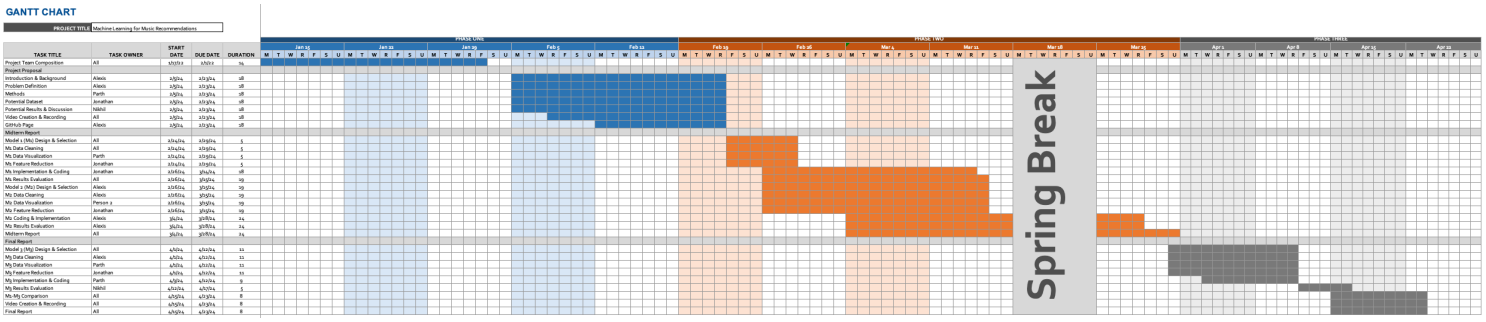
[3] F. Fessahaye et al., "T-RECSYS: A novel music recommendation system using deep learning," in 2019 IEEE International Conference on Consumer Electronics (ICCE), 2019. doi: 10.1109/ICCE.2019.8662028

[4] Ò. Celma and P. Herrera, "A new approach to evaluating novel recommendations," in Proceedings of the 2008 ACM conference on Recommender systems, 2008. doi: 10.1145/1454008.1454038

[5] D. Kowald, M. Schedl, and E. Lex, "The unfairness of popularity bias in music recommendation: A reproducibility study," in Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, doi: 10.1007/978-3-030-45442-5_5

Contributions and Gantt Chart

	Contribution
Parth	GMM + results + PCA
Jonathan	Neural Net + PCA
Nikhyl	Discussion/results + visualzation
Zachary	Decision tree + random forest + PCA
Alexis	Github page + video presentation



back

ml_project is maintained by awilmot6.

This page was generated by GitHub Pages.