# mlgroupproject

## Introduction & Background:

Leveraging machine learning to classify various music genres has become increasingly important with the rise in consumer interest for music streaming platforms, in order to enhance important features including playlist curation and music recommendations. Until now, various methods including support vector machines and deep learning [1], [2] have been used to explore this task, and there has also been novel work done using Residual Attention Networks [3]. Tzanetakis and Cook's work in this field introduced the GTZAN dataset which provides low level audio data for genre classification. Along with this the FMA dataset and Million Song dataset will be utilized for this project to provide over 10,000 songs labeled by genre with low-level audio data to help capture important details to distinguish between genres.

### Links to Datasets:

1. GTZAN: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification/code
2. FMA: https://github.com/mdeff/fma
3. Million Song Dataset: http://millionsongdataset.com/

## Problem Definition:

The primary goal of this project is to develop an effective model that can sort songs into different genres. We intend to identify specific audio features and qualities such as rhythm, volume, and utilized instruments that can contribute to a genre.

## Motivation:

The motivation behind this idea stems from the craze regarding personalized algorithms and music algorithms specifically. Curiosity with the inner workings of these algorithms birthed the initial steps of this project. We felt that discovery of music similar to music we enjoy is an aspect many modern streaming services lack. Hence, the motivation behind this project stems from the growing need for better music recommendation systems.

# Methods:

The three methods that we decided could be useful in preprocessing our data are PCA, DBSCAN and changing categorical data into numerical data. PCA will help us reduce the dimensionality of our data, so we don't use unnecessary features. DBSCAN will eliminate some of the noise of our data to make it more easily categorized. Changing numerical data into numerical data will ensure that it is compatible with our algorithms.

The three algorithms that we identified for our supervised and unsupervised learning are k means, logistic regression and neural networks. K means will be helpful in initially dividing our data into clusters representing different genres of music. Logical regression can be used to analyze the likelihood of a piece of music falling into a given genre. And finally a neural network can be trained to learn how to properly categorize inputted music.

For our midterm checkpoint, we chose Principal Component Analysis (PCA) for data processing and K-means clustering as our machine learning algorithm. Our primary objective was to begin organizing songs into clusters that could represent different music genres. Due to the large dataset we selected, we wanted to utilize PCA to reduce dimensionality and have it provide an initial evaluation of data. By visualizing the product after PCA was run on it, we hoped to identify certain patterns or aspects of the data that were obvious immediately. We believed this initial piece of information was vital and chose PCA as our pre-processing method to be implemented. For the first clustering algorithm we chose to implement, we went with k-means clustering. We felt that since it was one of the learning algorithms we had studied recently in class and were familiar with from implementing it in homeworks, it would represent a good and comfortable starting point for our project, and that we could expand off of it for the future based on the results. K-means is a fast algorithm and looks to provide strong relationships between data points in clusters, but it typically works best with data sets that do not have a lot of overlapping points. If we found this to be the case for our data, we could proceed with other forms of clustering algorithms that would be more suitable.

Expanding beyond the initial work we did for our midterm checkpoint, we decided to implement DBSCAN as an additional preprocessing method to help clean up our data. DBSCAN is an unsupervised clustering algorithm that groups points based on their proximity to each other, and also is able to identify noise points, which are outliers that are not close enough to other points to be a part of any cluster. Our dataset is quite dense, meaning DBSCAN would likely not be well-suited for genre identification, but we thought it would be helpful for removing noise and outlier points that could throw off our chosen supervised clustering algorithms. The additional clustering algorithms we chose to implement after doing k-means for our midterm checkpoint were Logistic Regression and Random Forest. After discussions with our mentor about the poor performance of our k-means model and its low fm score, we landed on these two algorithms as potentially better performing. Logistic Regression is good for data sets with fewer, more independent features,

where the relationship between the features and outcome is fairly straightforward. Random Forest on the other hand, with its use of bootstrap aggregation with multiple decision trees, handles more complex, higher-dimensionality datasets better. By implementing both, we were curious to see which would be better suited in handling our dataset, though due to the dimensionality and complexity of the music data we suspected Random Forest would be more effective.

Finally, after we had implemented and reviewed everything so far, we made the decision to implement a Neural Network as well. This is definitely the most complex model out of our selections, and its unique strength with its backpropagation across nodes and layers is its versatility and its ability to model very large, complex, and convoluted data sets. These qualities gave us high hopes that it would perform the best out of all the algorithms.

# Results and Discussion:

Three metrics that could be used to evaluate to explore the results of our algorithms are the fm score, correlation matrix and a precision recall curve. The fm score will give us a sense of how accurately our models can predict the genre of a piece of music. The correlation matrix will give us a sense of how different features of our musical data interact with each other. And finally the precision recall curve will give us the opportunity to visualize the accuracy of our model.

Our goal for this project is to have a model that can consistently and accurately categorize music into different genres. We will know our project can achieve this goal if we have an fm score that is close to 1 and a precision recall curve with an area under the curve that is close to 1. Additionally, having values in the correlation matrix close to 1 or -1 will help us interpret the correlation between different features.

We used PCA to reduce the dimensionality of our dataset, which contained 58 features, producing points with 21 components each and retaining 90% of the variance. This dimensionality reduction helped condense the features down to 21 components, simplifying the data while preserving critical information. By doing so, we aimed to decrease computational load and highlight the most informative aspects of the dataset, making it more manageable for clustering. Both of these are plotted using their two strongest components.

Midterm Images: https://drive.google.com/drive/folders/1DP2dCQdpQY4S9pOZXZ9Dm-pGeY5lbRVI?usp=drive_link (Link to the Visualizations / Images)

Figure 1 is what the points look like using their true labels, colored differently for each genre. Figure 2 is a k-means clustering of the data points using 10 clusters. Colors represent clusters, shapes represent original labels. The goal was to identify natural groupings within the music data that might correlate with genre distinctions. PCA's initial visualization of the data helped us examine patterns based on true genre labels, while K-means provided a clustering view. Despite

our efforts, the K-means model produced an fm score of 0.271, indicating it may not be the most effective algorithm for this task. The next steps will involve exploring more sophisticated models to improve genre classification accuracy. We believe Convoluted Neural Networks, DBSCAN and Logistic regression could achieve this. DBSCAN will help us get rid of outliers and capture patterns that are non circular, which is what k means is limited to. Convoluted Neural Networks and Logistic Regression on the other hand will both introduce supervised learning into our categorization, and hopefully allow us to build a more accurate model.

Adding onto PCA and K-means from our midterm checkpoint, we decided to implement DBSCAN, logistic regression, and random forest.

Firstly, DBSCAN's unsupervised clustering, using an epsilon of 0.9 and minpoints of 5, resulted in only 2 groups, one with the bulk of the points with 9950, and a second much smaller one with 7 points. This shows its unsuitability as a classification algorithm. However, it did identify 33 noise points that were not close enough to any cluster, which we were able to discard as outliers to hopefully improve the quality of our data. We also modified our PCA slightly, by adjusting the preserved variance to 0.98, which now reduces the data to have 43 components from the original 58. Being more strict with preserved variance does mean our dataset was not reduced in dimensionality as much, but we hoped our chosen algorithms would be effective even with a more complex dataset.

We also decided as a metric to utilize an fm score, which will give us a sense of how accurately our models can predict the genre of a piece of music. From the images below, it can be seen that all of the genres of music are mixed and scattered about in the original graph. When it went through k-means, it became somewhat categorized. As seen below, jazz (orange colored dots) and blues (blue colored dots) are mostly grouped together and in the same location on the K-means graph. However, the K-means model produced an fm score of 0.266, indicating poor clustering, as the predicted clusters are quite different from the true clusters. In addition, it indicates a low precision and recall which means it's assigning dissimilar points to the same cluster and failing to group similar ones together. Overall, it reinforces that K-means isn't the best algorithm for this task.

For the newly implemented logistic regression, it can be seen that the graph is somewhat grouped visually as compared to the original graph. The logistic regression model produced an fm score of 0.439 which is 0.173 higher than the fm score of k-means. This score indicates a moderate performance in this classification task. As a result, the precision and recall are also moderate which means it's sometimes assigning dissimilar points to the same cluster and sometimes failing to group similar ones together. Although it can be deemed as an acceptable score, there is also room for improvement.

For the newly implemented random forest, the organization of the points looks similar to the one in logistic regression just with different colors. The random forest model produced an fm score of

0.65 which is 0.211 more than the logistic regression fm score. Similar to logistic regression, this score also indicated a moderate performance but is better. It suggests that the model has a balanced but imperfect combination of precision and recall. It is still making errors when grouping dissimilar points and true positive pairs so there is still room for improvement.

Another metric that we decided to use were confusion matrices, providing a detailed piece of information on how well the model's predictions match the actual outcomes. For the logistic regression confusion matrix, it can be seen that the number of correct predictions (along the diagonal) amounted to 622 / 1000. As a result, it has an accuracy of 62.2%. This shows that our model is better than normal at predicting, but still has room for improvement. In contrast, the random forest confusion matrix produced 800 / 1000 correct predictions. As a result, it has an accuracy of 80% which indicates that our model is capturing useful patterns from the data.

Building upon these results, the neural network model emerged as the best-performing method. The neural network achieved an FM score of 0.776, surpassing both logistic regression and random forest. Visually, the points were more clearly separated into genres, reflecting the model's ability to learn and differentiate complex patterns. Its confusion matrix demonstrated 811 correct predictions out of 1000, resulting in an accuracy of 81.1%. Furthermore, the precision-recall (PR) curves highlighted the model's high average precision (AP) scores, particularly for blues, classical, and metal genres. This strong performance underscores the flexibility of neural networks in modeling intricate relationships within the data.

The confusion matrices can be leveraged alongside precision recall curves to understand which categories different models excel at or struggle at. For both Logistic Regression and Random Forest, the worst precision recall curve belonged to rock. When examining the confusion matrix, it can be seen that in both models rock was predicted currently a very small percent of the time, in Logistic Regression, only guessing rock currently 7 out of 100 times. Other problem categories were pop, disco and reggae, with both Logistic Regression and Random Forest tending to over-predict pop and reggae, and commonly confuse disco with pop, underpredicting that category. Neural Networks were able to tighten the classification of all of these categories across the board, improving the overall performance of the mode.

Despite its strong performance, the neural network is not without tradeoffs. It requires significantly more computational resources than both logistic regression and random forest, making it less practical for real-time or resource-constrained environments. Additionally, neural networks can be more challenging to interpret, as their decision-making process is not as transparent as tree-based models. Regularization techniques, such as dropout and L2 regularization, are also essential to prevent overfitting, especially when working with smaller datasets. In our specific case, it can be said that random forest is performing better at this classification task compared to K-means and logistic regression. This could be due to the more flexible nature of random forests but also comes with drawbacks such as a higher computational cost. However, the neural network takes this flexibility further, capturing nuances in the data that

other models miss. While it performs exceptionally well, its higher complexity and training time highlight the need for careful consideration depending on the application's requirements. The tradeoffs between these methods are clear: logistic regression provides a simple, interpretable model but struggles with non-linear relationships. Random forest bridges the gap, offering improved performance while retaining some interpretability. Neural networks deliver the best results but demand more resources and careful tuning. K-means and DBSCAN, as unsupervised methods, are less suited for this task but DBSCAN contributed to data preprocessing by identifying clusters and noise.

Final Images: https://drive.google.com/drive/folders/1RwJkbwCxAY3OFJoAyAyxi4rPpckaHALd?usp=sharing (Link To Visualizations / Images)

# References:

https://link.springer.com/article/10.1007/s00500-023-08093-0 [1] Y. Xia, "Impact of AI-assisted music classification in video games for sustaining effectiveness," Soft Computing, Apr. 2023, doi: https://doi.org/10.1007/s00500-023-08093-0.

https://iopscience.iop.org/article/10.1088/1755-1315/785/1/012013/meta [2] R. Anand, R. S. Sabeenian, D. Gurang, R. Kirthika, and S. Rubeena, "AI based Music Recommendation system using Deep Learning Algorithms," IOP Conference Series: Earth and Environmental Science, vol. 785, no. 1, p. 012013, Jun. 2021, doi: https://doi.org/10.1088/1755-1315/785/1/012013.

https://ieeexplore.ieee.org/abstract/document/8823100 [3] Q. H. Nguyen et al., "Music Genre Classification using Residual Attention Network," IEEE Xplore, Jul. 01, 2019, doi: 10.1109/ICSSE.2019.8823100.

# Gantt Chart:

| **GANTT CHART** | | | | | | | |
|---|---|---|---|---|---|---|---|
| PROJECT TITLE | Music Genre Categorizer | | | | | | |
| | | | | | PHASE ONE | | |
| TASK TITLE | TASK OWNER | START DATE | DUE DATE | DURATION | Sep 27 | | |

| GANTT CHART | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | M | T | W |
| Project Proposal | | | | | | | |
| Introduction & Background | Aniketh & Jason | 9/27/2024 | 10/4/2024 | 7 | | | |
| Problem Definition | Krishnav & Ivan | 9/27/2024 | 10/4/2024 | 7 | | | |
| Methods | Reid & Krishnav | 9/27/2024 | 10/4/2024 | 7 | | | |
| Potential Results & Discussion | Reid | 9/27/2024 | 10/4/2024 | 7 | | | |
| Video Recording | Ivan | 10/4/2024 | 10/7/2024 | 3 | | | |
| GitHub Page | Aniketh | 10/4/2024 | 10/7/2024 | 3 | | | |
| Model 1 | | | | | | | |
| Data Sourcing and Cleaning | Aniketh | 10/7/2024 | 10/15/2024 | 8 | | | |
| Model Selection | Ivan | 10/15/2024 | 10/18/2024 | 3 | | | |
| Data Pre-Processing | Jason | 10/18/2024 | 10/25/2024 | 7 | | | |
| Model Coding | Krishnav | 10/25/2024 | 11/8/2024 | 13 | | | |
| Results Evaluation and Analysis | Reid | 11/8/2024 | 11/16/2024 | 8 | | | |

| GANTT CHART | | | | | | | |
|---|---|---|---|---|---|---|---|
| Midterm Report | All | 11/8/2024 | 11/16/2024 | 8 | | | |
| Model 2 | | | | | | | |
| Data Sourcing and Cleaning | Aniketh | 10/18/2024 | 10/22/2024 | 4 | | | |
| Model Selection | Jason | 10/22/2024 | 10/25/2024 | 3 | | | |
| Data Pre-Processing | Reid | 10/25/2024 | 10/29/2024 | 4 | | | |
| Model Coding | Krishnav | 10/25/2024 | 11/19/2024 | 24 | | | |
| Results Evaluation and Analysis | Ivan | 11/19/2024 | 11/24/2024 | 5 | | | |
| Model 3 | | | | | | | |
| Data Sourcing and Cleaning | Ivan | 10/18/2024 | 10/22/2024 | 4 | | | |
| Model Selection | Krishnav | 10/22/2024 | 10/25/2024 | 3 | | | |
| Data Pre-Processing | Reid | 10/25/2024 | 10/29/2024 | 4 | | | |
| Model Coding | Jason | 10/25/2024 | 11/19/2024 | 24 | | | |
| Results Evaluation and Analysis | Aniketh | 11/19/2024 | 11/24/2024 | 5 | | | |

| GANTT CHART | | | | | | | |
|---|---|---|---|---|---|---|---|
| Evaluation | | | | | | | |
| Model Comparison | Jason | 11/29/2024 | 12/7/2024 | 8 | | | |
| Presentation | All | 11/29/2024 | 12/6/2024 | 7 | | | |
| Recording | All | 12/6/2024 | 12/7/2024 | 1 | | | |
| Final Report | All | 11/29/2024 | 12/7/2024 | 8 | | | |

# Contribution Table:

| Member | Contributions (Milestone 1) | Contributions (Milestone 2) | Contributions (Milestone 3) |
|---|---|---|---|
| Aniketh | Set up github pages, wrote introduction, video | Updating Github Pages, Editing the Midterm Checkpoint Report | Updating Github Pages, Final Presentation, Video Recording |
| Ivan | wrote video slides, wrote problem definition and motivation, video | Coding, Editing the Midterm Checkpoint Report, Gantt Chart | Final Report - Results & Discussion |
| Reid | Wrote methods, wrote results and discussion, video | Coding the ML Model (PCA and kmeans), Gantt Chart | Coding the ML models, Final Report |
| Jason | Research and citations, introduction, video | Implemented solution, Results and Discussion Midterm Report | Final Report - Results & Discussion, Methods |
| Krishnav | problem definition and description, video recording | Implemented solution, Results and Discussion Midterm Report, Contribution Table | Final Report - Results & Discussions |