Music Genre Classification

Akshat Nitin Savla
North Carolina State University
asavla@ncsu.edu

Praneya Nimesh Lal
North Carolina State University
plal@ncsu.edu

Atharv Prashant Pandit
North Carolina State University
apandit3@ncsu.edu

1 INTRODUCTION AND BACKGROUND

In this section, we will outline our project's problem description and related activity.

1.1 Problem Statement

Understanding music has been a challenging task for many experts for a long period of time, in that, classifying it into different type of genres is more arduous. The major reason for choosing this project was to learn how to recognize songs with varied tones, harmonies, and sound imagery. Most genres have similar characteristics, and as the number of various genres and musicians grows, it becomes more difficult to separate music. To address this issue, we employed the GTZAN Music Genre Classification dataset in our project, which has been widely used for music genre categorization. The dataset includes an audio and images folder, as well as two CSV files. For this project, we're using the genres original folder, which has 10 genres with 100 audio files each, and the librosa module in Python for feature extraction. After extracting spectral and rhythm information, our baseline KNN model is trained and tested to produce the desired output.

There are numerous uses for this technology, such as individualized music suggestion and classification for each user. Most of the genres, differ slightly based on specific features and then the streaming platforms classifies them accordingly. For music recommendation, we can group users with similar choice and suggest songs based on that to different types of music listeners. Hence, there are many applications but the major purpose of this project is to correctly classify audio files into their various genres and distinguish music genres based on tone, pitch, melody, and rhythm.

1.2 Related Work

[1] discusses a method for music genre recognition using deep learning with KNN. It works by first distinguishing the linguistic content and removing the noise before feeding it into the model. This step is crucial because it can help to remove irrelevant features and noise from the audio signal, allowing the model to focus on more important features that are relevant to the classification task.

[2] discusses how music information retrieval has been a challenging task and music classification is an very important part of it. Distinguishing between different genres of music is a difficult task as they all seem to be similar to certain extent. Signal processing is done for feature extraction and various machine learning models were selected, trained and compared for genre classification on the GTZAN dataset. Various classifiers like logistic regression, K-Nearest Neighbors, Random Forest Classifier, Support Vector Machine and XGBoost were used for music genre classification. Then Convolutional Neural Network and other ensemble methods were used to classify and identify different types of music genres. Ensemble methods are the methods that combine various models and then their output is combined to get the final result and continuous iterations results in improvement of the desired result. Best results were observed using Convolutional Neural Networks in [2], various parameters and graphs were presented for properly analyzing the data and hyper-parameter optimization was done to continuously improve the model performance and best model is selected for classification.

Deep Learning methods using convolutional neural nets are are also effective for recognizing the genre of music. [3] utilizes spectrograms generated from time-slices of songs as the input into a neural network to classify the songs into their respective musical genres. First, using spectrograms as input allows us to capture the frequency and temporal features of the audio signal, which are critical for distinguishing between different genres of music. Second, using a neural network allows us to learn complex patterns and relationships between the input data and the output labels, potentially improving the accuracy of the classification. [4] also uses spectograms to train various artificial neural nets, and subsequently comparing the performance of the models. The authors were successfully able to significantly improve the performance than existing spectogram based models.

Other machine learning methods like Support Machine Vectors (SVM) are also used for the music genre classification task at hand. [5] compares the effectiveness of KNN and SVM models for predicting the genre of songs using MFCC features. The research also explores the use of dimensionality reduction via principal component analysis (PCA) on the MFCC features. This is similar to the proposed approach which also involves feature reduction techniques such as PCA and selecting the most relevant features for classification.

2 PROPOSED METHOD

In this section, we will discuss our proposed methodology and then provide rationale for the method we have chosen.

2.1 Novel Aspects

(1) A novelty in our project was the creation of an appropriate dataset to be fed into a machine learning model. The dataset used comprised of music files in .wav format. These audio files constitute unstructured data which cannot be used for training models. Hence, we used feature extraction to extract various auditory features from the audio files. We converted the unstructured data into structured numerical data. Subsequently, we used this data to train our machine learning models. We also removed the files that were erroneous, and used the other files for properly classifying the different music genres.

(2) The second novelty of our project is the use of manifold learning. Manifold learning is a kind of non-linear dimensionality reduction technique. Essentially it attempts to discover the underlying manifold by finding a mapping from the high-dimensional data space to a lower-dimensional space that preserves the local geometric relationships between the data points. Some algorithms for manifold learning include Locally Linear Embedding (LLE), t-SNE, and Uniform Manifold Approximation and Projection (UMAP). For this project we used t-SNE and have tried to improve model performance by performing dimensionality reduction by extracting non-linear features from our dataset. We have described the methodology and results of manifold learning furthwe in the report.

2.2 Approach

Initially, we had audio files that were 30 seconds long and belonged to one of ten categories. Each of these audio recordings is divided into ten 3 second files. This is done so that our Machine Learning algorithms can learn features from more samples of each class. The library-librosa was then utilized to extract various audio features for our analysis. This is one of the novelties since we extracted features from audio files that allow us to use machine learning algorithms on our data. The majority of our characteristics are derived from MFCCs, or Mel Frequency Cepstral Coefficients. These are widely used for representing special characteristics of audio signal in a compact and efficient way. We find the mean and variance of each of these components across the samples in an audio file. We also extract other relevant features[4] such as tempo, rms (loudness), etc. Once we have the extracted features, we perform feature scaling since most of the Machine Learning algorithms are sensitive to scaling. A correlation matrix is also used to determine whether any of the features are highly connected and to do feature selection. We also execute and report on dimensionality reduction using PCA and Manifold Learning. One of the project's peculiarities is the use of manifold learning to learn nonlinear characteristics.

KNN acts as a great baseline model for our other models as it sets a threshold value, which can be compared to the performance of our other models and their aim would be to perform better than our baseline model. Firstly we used Support-Vector Machines for classifying the various types of music genres. Linear and Non-Linear decision boundaries are effectively managed by SVM, then we tried different kernel functions like polynomial, linear, rbf and sigmoid. We tried all the different functions so that we can compare and select the kernel which gives the highest classification accuracy. Secondly we used Random-Forest classifier, which is an ensemble learning method in which multiple decision tress results are combined and final classification of the music genre is done. Lastly, we used manifold learning to understand the structure of our data properly and convert high-dimensional data into a low dimensional space, it is believed that this helps in increasing the classification accuracy of our model. We took the best metric as accuracy to compare all our above models and then choose the model with the highest accuracy. Best accuracy is given by SVM for doing music genre classification using the GTZAN dataset.

2.3 Rationale

We extracted features from the audio files because audio files are unstructured and are not suitable for machine learning models. Hence, we extract various auditory features from the audio files, so that we can generate tabular data consisting of numbers, which is suitable for machine learning models. We divide each audio file into 10 files of 3 seconds since we only have 100 files in each class. This increases the probability of underfitting due to minimal data. Hence, we perform data augmentation. Data augmentation is the process in which we increase the sample size of the dataset so that there is much variability in it and new samples are created from the existing dataset, so that our model is robust and can handle all the edge cases properly, it helps in improving generalization. We perform feature selection using correlation coefficient matrix, as well as create models after dimensionality reduction such as PCA and manifold learning to avoid the curse of dimensionality. Manifold Learning is basically mapping the high-dimensional data to low dimensional place.

Our choice of using KNN as a baseline is justified since KNNs are simple models that make a good choice for basic classifiers. Then we used Support-Vector Machine, Random-Forest Classifier and train models to improve the performance over our baseline models. SVMs work well for this domain as they handle high-dimensional data pretty well and also capture complex non-linear relationships that exist in the data. All the models are trained and different music genres are classified using them, then we check our results. The durability of Random Forest to noise and its aptitude for handling imbalanced datasets are well recognized.

3 EXPERIMENT

This part will go over the dataset, hypothesis for our project, and experimental design.

3.1 Dataset

The project aims to classify the genres of music into different types, we selected the GTZAN dataset [6] which is available on Kaggle for this purpose. The main inspiration for this project was to understand the difference between different tones in music and visualize sound. This dataset is often used for Music Genre Recognition; the files were collected in the period of 2000-2001, from a variety of sources such as CDs, radio and microphone recordings.

The data was collected under various recording situations to ensure that the data is diverse and can accurately represent the majority of real-world occurrences. This is an excellent practice because evaluating and working on this dataset will help us discover new trends and gain a thorough understanding of music genre classification. The dataset's diversity would aid in training our machine learning model, ensuring that the model does not overfit to the training data.

There are two folders in this dataset: genres original and images original, along with this there are 2 CSV files. The genres original folder has ten different music genres, each with 100 audio files that are 30 seconds long. The images original folder contain the visual representation of data for each genre. For this project we are primarily using genres original folder for training and testing

our model. The audio clips in the dataset are in WAV format, and the images are in PNG format. The dataset is 1GB in size. We will extract the features, split the data into training and testing sets, and then train our machine learning model using various techniques and hyper-parameters to effectively predict and classify the many sorts of music genres.

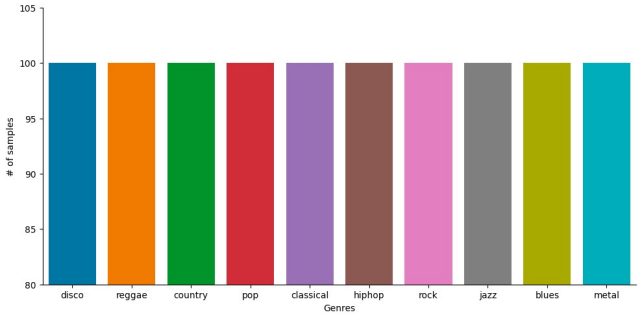

Figure 1: Original Class Distribution of Music Genres

3.2 Hypothesis

- Selecting features using feature selection techniques like correlation matrix, PCA, manifold learning. We will compare how our models perform before and after feature selection.
- The length of audio samples can impact the accuracy of music genre classification. The original dataset contains audio samples of length 30 seconds. The performance of the model can be tested by using samples of smaller lengths by splitting the audio files into multiple smaller files.
- The choice of classifier can affect the performance of music genre classification. Using various machine learning methods like KNN and SVM and deep learning models like ANN and CNN can affect the performance of recognition task.

3.3 Experimental Design

To assess the viability of the above hypotheses, we must consider all of the essential steps. The stages are as follows:

- **Data Cleaning and Feature Extraction**
  Data cleaning involved removing one file from the dataset since it was corrupted. First, we divided each audio clip of 30 seconds into ten 3 second audio files. This was done to make the data-set larger. Figure2 shows the class distribution after performing the above steps. We will also train models on the original 30 second audio clips to verify our hypothesis, if the length of a song has any effect on the accuracy of the model. We extracted different features from the audio files using the librosa[7], a python library for audio analysis. We store these extracted features in a csv file. This was one of the novelties of our project, where we created a new csv dataset from a dataset of audio files. The extracted features were:
  - 20 MFCC mean values for each sample in the audio file
  - 20 MFCC variance values for each sample in the audio file
  - Spectral Centroid Mean
  - Spectral Centroid Variance
  - Spectral Rolloff Mean
  - Spectral Rolloff Variance
  - Spectral bandwidth
  - Tempo
  - RMS Mean (Measure of Loudness)

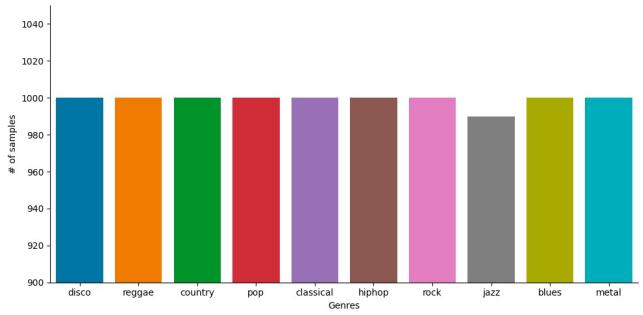

Figure 2: Class Distribution after Splitting the Audio Files

Feature scaling is applied to the data since most machine learning models are sensitive to scaling. For example, if the data has features with different ranges, the Eucledian distance will be dominated by the attribute that has high values. This makes feature scaling necessary. For this reason, we applied standardization to the data.

- **Feature Selection (Data Pre-Processing)** We use different feature selection techniques such as eliminating correlated features, Principal Component Analysis (PCA), and Manifold Learning to reduce the number of dimensions. We train our models with and without applying these techniques and compare results.

(1) Correlation
We perform feature selection on the 47 features using the correlation matrix in Figure 3. We remove features which have a correlation coefficient > 0.9 with the other attributes. The summarized results of the correlation matrix using a heatmap are shown in Figure3. We remove the following 3 features:'spectral bandwidth','spec_centroid_mean', and 'spec_rolloff_mean'.

(2) PCA
Principal Component Analysis (PCA) is performed on our dataset for dimensionality reduction. Figure 4 shows the results of performing PCA. We used 30 as the number of principal components and trained different models for classification.

(3) Manifold Learning
We also tried Manifold Learning, a non-linear dimensionality reduction technique. To get the optimal number of components for the number of dimensions of our data that the manifold learning should return, we use GridSearchCV on the

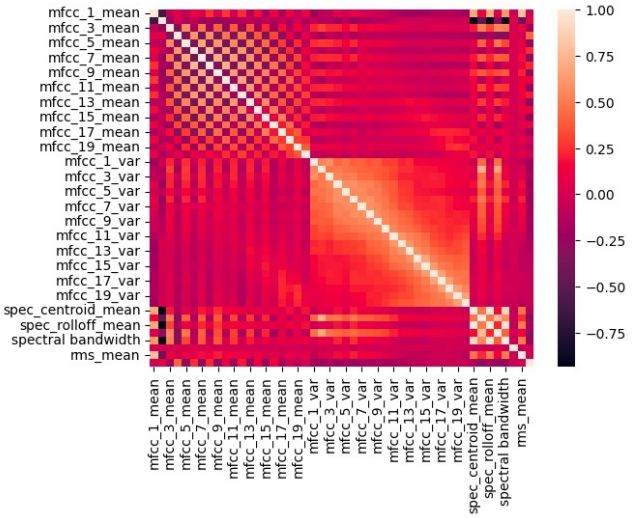

Figure 3: Heatmap of Correlation Coefficient

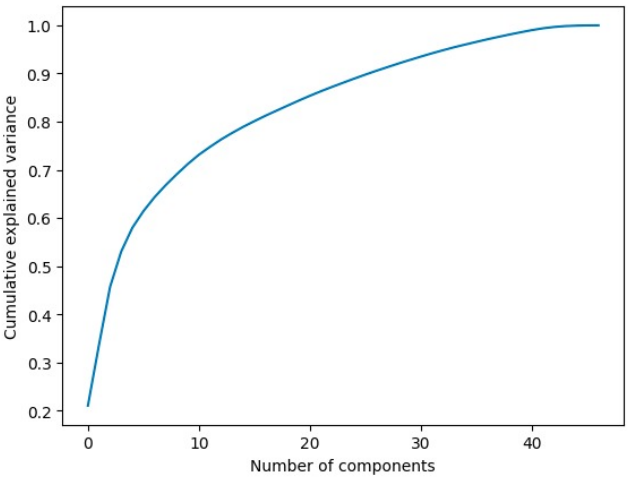

Figure 4: Cumulative Variance and # of PCA components

number of components [2,5,10,15,20] using manifold learning. We perform the GridSearchCV with the other model parameters as well.

- **Model Training and Selection** We have trained different types of classifiers and used a KNN classifier as a baseline model. We also try other classifiers such as well such as SVMs and Random Forests. We will then compare the accuracies of the different models on the dataset.
  In our music genre classification task using the GTZAN dataset, we employed cross-validation (CV) to evaluate the performance of our models and to tune the hyperparameters. For the KNN model, we used a 10-fold cross-validation approach, while for the Random Forest and SVM models, we used a 5-fold cross-validation approach.

  To find the optimum hyperparameters for our model, we used the grid search method, and implemented it using scikit-learn python library's GridSearchCV function. For KNN the hyperparameters that we tuned was k, the number of nearest neighbours. For SVMs, the hyperparameters that we tuned were the kernel function used, C, and n (for kernel = rbf). For Random forest, the hyperparameters tuned were the maximum depth and the number of decision stumps used.
  The evaluation metric used to evaluate the performance of the models was accuracy. In this context, accuracy measures the percentage of correctly classified tracks out of the total number of tracks in the dataset.

4 RESULTS

In this section we will explain the results we got after training and testing our model. We experimented with KNNs to get a baseline for our final model.

4.1 KNN Model

(1) To begin, we utilize the entire dataset with 47 features to train the model. We employed KNNs with different nearest neighbor values and obtained the cross-validation scores as shown in Table 1.
Hence, we selected nearest neighbors = 3 and achieved a Test Accuracy of 84%.

(2) We perform feature selection using the correlation coefficient matrix. Our cross validation results indicate that a nearest neighbors = 3 yield the best results.
A test accuracy of 83% is achieved when we use a KNN model with nearest neighbors = 3. Hence, we see that the feature selection has not made a drastic improvement on our model.

(3) Principal Component Analysis (PCA) is performed on our dataset for dimensionality reduction. Figure 4 shows the results of performing PCA. We used 30 as the number of principal components, and used KNN for classification. On training the KNN model with nearest neighbors=3, we achieved a test accuracy of 81%.
We can observe that feature selection and dimensionality reduction have had no effect on the performance of our KNN models.

Table 1: Cross Validation Scores for KNN models

	Cross Validation Score		
# of neighbours	**Initial**	**After feature selection**	**After PCA**
3	83%	82%	79%
5	79%	78%	75%
7	75%	74%	71%

(4) For Manifold learning, we use a KNN model with n=3 and try different number of components returned by the dimensionality reduction performed by the manifold learning. We have the number of components as a hyper-parameter. We see that we

get the best validation accuracy with number of components=3. We get a test accuracy of 84% Hence, we observe that the Mani-

Table 2: Manifold Learning Cross Validation Scores

# of Components	Cross Validation
1	0.59
2	0.83
3	0.84

fold learning has not had a positive effect on our model.

4.2 Support Vector Machine (SVM)

For SVMs we tune the hyperparameters like-C (regularization parameter), kernels and different degrees for the polynomial kernel.

(1) We train our SVMs on the entire dataset. Our results are summarized in Table 3. We see that we get the highest cross-validation score with a C of 10 and a rbf kernel. We get a test accuracy of 89% with these hyperparameters.

Table 3: Performance of SVM model

Kernel \ C	0.1	0.5	1	3	5	10
linear	0.69	0.69	0.69	0.69	0.69	0.69
poly	0.61	0.71	0.75	0.81	0.82	0.84
rbf	0.65	0.76	0.8	0.85	0.85	**0.87**
sigmoid	0.54	0.5	0.47	0.45	0.44	0.44

(2) On performing feature selection using the correlation coefficient matrix. Our cross validation results indicate that a C of 10 and a rbf kernel yield the best results.
A test accuracy of 88% is achieved as seen in Table 4 when we use these hyperparameters. Hence, we see that the feature selection has not made a drastic improvement on our model.

Table 4: Performance of SVM model after feature selection

Kernel \ C	0.1	0.5	1	3	5	10
linear	0.68	0.68	0.68	0.68	0.68	0.68
poly	0.62	0.70	0.75	0.81	0.82	0.83
rbf	0.65	0.75	0.8	0.84	0.85	**0.86**
sigmoid	0.55	0.5	0.47	0.44	0.44	0.43

(3) We train our SVMs on the dataset after performing PCA and extracting 30 principle components. We see that the C of-10 and kernel-rbf give the highest cross validation score.
On training the SVM with these hyperparameters, we achieved a test accuracy of 84% as seen in Table 5.
We can observe that feature selection and dimensionality reduction have had no effect on the performance of our SVM models.

Table 5: Performance of SVM model after PCA

Kernel \ C	0.1	0.5	1	3	5	10
linear	0.64	0.64	0.64	0.64	0.64	0.64
poly	0.59	0.67	0.71	0.75	0.77	0.78
rbf	0.63	0.72	0.75	0.80	0.81	**0.82**
sigmoid	0.52	0.46	0.43	0.42	0.42	0.41

(4) For Manifold learning, we use a SVM model with C=10 and rbf kernel and try different number of components returned by the dimensionality reduction using manifold learning. . We see that we get the best validation accuracy with number of components= Hence, we observe that the Manifold learning

Table 6: Performance of SVM after Manifold Learning

# of Components	Cross Validation
1	0.43
2	0.58
3	0.64

has not had a positive effect on our model.

4.3 Random Forest Classifier

For Random Forests we tune the hyperparameters like-max_depth which checks overfitting, and the number of estimators which are the number of decision stumps whose results will be combined to get the final result.

(1) We train our Random Forests on the entire dataset. Our results are summarized in Table 7. We see that we get the highest cross-validation score with number of estimators=200 and a max-depth=20. We get a test accuracy of 82% with these hyperparameters.

Table 7: Performance of Random Forest

n_estimators/ max_depth	5	10	20	100	150	200
10	0.61	0.67	0.71	0.74	0.75	0.75
20	0.63	0.69	0.75	0.80	0.80	**0.81**
50	0.63	0.70	0.74	0.80	0.81	0.80
100	0.63	0.70	0.75	0.80	0.80	0.81

(2) On performing feature selection using the correlation coefficient matrix. Our cross validation results indicate that a number of estimators=200 and a max-depth=50 yield the best results.
A test accuracy of 84% is achieved as seen in Table 8 when we use these hyperparameters. Hence, we see that the feature selection has made an improvement on our model.
(3) We train our Random Forest on the dataset after performing PCA and extracting 30 principle components. We see that the number of estimators=200 and a max-depth=100 give the highest cross validation score.

Table 8: Performance of Random Forest after feature selection

n_estimators/ max_depth	5	10	20	100	150	200
10	0.62	0.67	0.71	0.74	0.75	0.75
20	0.61	0.69	0.75	0.80	0.81	0.81
50	0.61	0.69	0.75	0.80	0.81	**0.81**
100	0.61	0.70	0.74	0.80	0.80	0.81

On training the Random Forest with these hyperparameters, we achieved a test accuracy of 76% as seen in Table 9.
We can observe that feature selection and dimensionality reduction have had no positive effect on the performance of our SVM models.

Table 9: Performance of Random Forest after PCA

n_estimators/ max_depth	5	10	20	100	150	200
10	0.53	0.59	0.64	0.67	0.68	0.68
20	0.53	0.61	0.67	0.73	0.74	0.74
50	0.53	0.61	0.67	0.73	0.74	0.74
100	0.52	0.60	0.67	0.73	0.74	**0.74**

(4) For Manifold learning, we use a Random Forest model with max-depth=50 and number of estimators=200 and try different number of components returned by the dimensionality reduction using manifold learning. We see that we get the best validation accuracy with number of components= Hence, we

Table 10: Performance of Random Forest after Manifold Learning

# of Components	Cross Validation
1	0.59
2	0.86
3	0.86

observe that the Manifold learning has had a positive effect on our model.

4.4 Comparing Results of Different Length Samples

We developed our models using samples of length of 3 seconds. We will check if having different length samples have an effect on our model. For this, we compare the results we get for KNNs with samples of length 3 seconds and length 10 seconds. We get the following following results:
Hence we can conclude that using samples of a longer length does not have a positive effect on the performance of our model.

Table 11: Comparison of performance with samples of different length

Neighbors	Samples of length 3sec	Samples of length 10sec
3	83 %	76 %
5	79 %	67 %
7	75 %	62 %

4.5 Discussion

Based on the results mentioned above we can infer and interpret the following:

- In the results section we compared all the different models and got valuable insights regarding the accuracy and efficacy for music genre classification. We used accuracy as our primary metric to select our best model.
- Looking at the obtained results, we can evaluate our hypotheses. We performed feature selection by using techniques such as correlation matrix, Principle Component Analysis and Manifold learning. The features that were really significant were extracted and all the models were compared before and after applying the feature selection techniques.
  - The feature selection method using correlation matrix did not really improve the performance of our models. One possible explanation for this might be that the non-linear, complex features were not captured by the correlation matrix.
  - Principal Component Analysis also failed to improve the accuracy of most our models. Given that the number of principal components selected is most likely true as we used the elbow method the select the optimum number of principal components, a plausible explanation for this can be that the principal components did not capture enough variance in the data.
  - Manifold learning improved the performance of some of our models. It improved the performance of KNN and Random Forest models, while the performance of SVM went down slightly. The improvement in the performance can be attributed to the fact that manifold learning could capture the non-linear dependencies in the data.
- Another hypothesis was whether duration of the audio sample can have an impact on the music classification task. As the length of the samples increases it becomes difficult to classify them, so we are testing the audio files with smaller sample lengths to check how does our accuracy change with the reduction in sample size. After testing we observed that accuracy for smaller samples of 3 seconds is higher as compared to the original samples which are of 30 seconds in length. This can be because the total number of samples increased and hence the model got more data to learn from and ended up performing well.
- There are various classifiers that can used to perform music genre classification with its pros and cons which makes choosing a right classifier for music genre classification is necessary. Out of the models that we implemented, the SVM model outperformed the baseline KNN as well as the random forest models. The best SVM model had an accuracy of 82%.

- The papers discussed in the related work section make use of machine learning and deep learning techniques for music genre recognition. They propose different approaches to extract audio features and classify music into different genres, such as using feature selection techniques, various classifiers like KNN, SVM, logistic regression, random forest, and XGBoost, and deep learning models like CNN and ANN. Preprocessing the audio data is also discussed to improve classification accuracy, and this also helped us to improve our accuracy.

5 CONCLUSION

- The classifier used can have a substantial influence on classification task accuracy, and different classifiers have distinct strengths and drawbacks. Feature selection and engineering are essential components of the classification process and can have a significant influence on model performance.
- SVM performed the best with a test accuracy of 89%, while Random Forest performed closely well with manifold learning with the number of components = 2 with an accuracy of 86%.
- The amount and quality of the dataset has a major impact on the classification model's accuracy. While experimenting we observed that SVMs and random forests, were more effective than others for music genre classification. Feature selection and engineering, such as the use of Mel-frequency cepstral coefficients (MFCCs) and chroma characteristics, were critical for obtaining high accuracy.
- We also successfully implemented two novelties for our project. The first one being creation of a new dataset from an unstructured dataset, while the second one being implementing a new feature engineering technique called manifold learning.
- Finally, this effort has demonstrated the significance of carefully evaluating the many components of music genre classification, as well as insights into the strengths and limitations of various machine learning models. Future research might look into more advanced techniques, such as deep learning algorithms, to increase the accuracy of music genre classification.

6 MEETING ATTENDANCE

Date	Time	Agenda
April 7, 2023	3pm-4pm	SVM
April 14, 2023	3pm-4pm	Random Forest
April 17, 2023	3pm-4pm	Manifold Learning
April 19, 2023	4pm-5pm	Report Discussion
April 21, 2023	4pm-5pm	Update on Progress

All three group members were present for the meetings and actively participated in all the discussions.

REFERENCES

[1] D. K. A. K. M. K. Dr. S. Ponlatha, Mathisalini B, "Music genre classification using deep learning with knn," in *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 12, no. 2, 2021, pp. 224–230.

[2] V. Shah, A. Tandle, N. Sharma, and V. Sheth, "Genre based music classification using machine learning and convolutional neural networks," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021, pp. 1–8.

[3] N. Pelchat and C. M. Gelowitz, "Neural network music genre classification," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 2019, pp. 1–4.

[4] G. G. A. N. Navneet Parab, Shikta Das, "Music genre classification using deep learning," in *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 10, 2021, pp. 1459–1467.

[5] M. Ali and Z. Ahmed, "Automatic music genres classification using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 8, 01 2017.

[6] Andrada, "GTZAN Dataset - Music Genre Classification." [Online]. Available: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[7] "Librosa." [Online]. Available: https://librosa.org/doc/latest/index.html