# CSE342: SML - Music Genre Classification

Mehul Agarwal
*Computer Science & Engineering Dept.*
*IIIT-Delhi, India*
mehul22294@iiitd.ac.in

Rahul Omalur Ramesh
*Computer Science & Engineering Dept.*
*IIIT-Delhi, India*
rahul22392@iiitd.ac.in

***Abstract*—This report presents the findings from a project aimed at developing a Music Genre Classifier utilizing various machine learning and deep learning techniques. The classifier's purpose is to streamline radio stations and genre-based playlists, analyze music for industry insights, organize content efficiently, aid producers and musicians in audio and music production, as well as assist in music licensing and copyrights. Models based on these algorithms are trained and compared for accuracy using the 'GTZAN genre collection' dataset, which comprises 10 genres with 100 samples each. Index Terms - K-Nearest Neighbors (KNN), Logistic Regression, Artificial Neural Network (ANN), Convolutional Neural Networks (CNN), Convolution-Recurrent Neural Network (CRNN).**

## I. Introduction

In a world where musical content is being pushed out at an exponential rate, with a wide variety of genres, it is ideal to create a Music Genre Classifier for purposes such as:

1) Streamlining Radio Stations and Genre-based Playlists
2) Music Analysis and Industry Insights
3) Efficient Content Organization
4) Aiding producers and musicians in Audio and Music Production
5) Music Licensing and Copyrights

Music genre classification is the task of automatically assigning a label to a piece of music based on its genre, such as rock, jazz, or classical. This task is challenging because genres are often subjective, ambiguous, and overlapping, and there is no clear definition of what constitutes a genre. Moreover, different music genres may share similar acoustic features, such as tempo, rhythm, or instrumentation, making it hard to distinguish them based on audio signals alone.

## II. Dataset

We utilize two datasets for training and testing our models which are obtained by the methods that will be explained in the following 'Data Pre-processing' section . These datasets were extracted from the main GTZAN dataset, which is a widely used collection of audio files organized into 10 genre categories. It is the main music sample dataset that is used in music genre classification research.

*CSE342: Statistical Machine Learning (Winter 2024), A V Subramanyam, IIIT-Delhi.*

## III. Data Pre-Processing

The audio files were preprocessed to extract various audio features such as tempo, spectral centroid, and zero-crossing rate. These features were then saved into CSV files, `features_30_sec.csv` containing features aggregated over 30-second segments and `features_3_sec.csv` containing features aggregated over 3-second segments. These CSV files serve as input features for our classification models.

The transformation of audio data into visual representations known as mel-spectrograms, which are useful for machine learning tasks, facilitates the conversion of raw audio files into images, enhancing the accessibility and interpretability of the data. This preprocessing step prepares the data for analysis and model training, contributing to the effectiveness of audio-based machine learning applications.

`extract_melspectrogram_features`: This function takes audio files as input, transforms them into mel-spectrograms (visual representations), and presumably saves these mel-spectrograms as images.

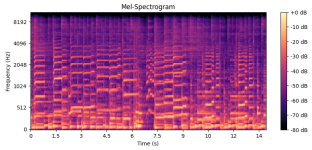
## IV. Literature Review

Music genre classification is a fundamental task in music information retrieval, with various machine learning and deep learning algorithms being employed to achieve accurate classification results. In this section, we provide an overview of some of the commonly used algorithms for music genre classification and some common terminologies in sound processing.

### A. Common Terminologies in Sound Processing

- **Spectrogram**: A visual representation of the frequency spectrum of a signal over time. It displays the distribution of frequencies present in the signal at different time points. Spectrograms are widely used in tasks such as speech recognition, music analysis, and sound classification.
- **Mel-frequency Cepstral Coefficients (MFCCs)**: Features extracted from the short-term power spectrum of an audio signal. MFCCs capture both spectral and temporal characteristics and are commonly used in tasks such as speech and speaker recognition, music genre classification, and audio fingerprinting.
- **Mel-Spectrogram**: A spectrogram with the frequency axis converted to the mel scale, which is perceptually uniform. Mel-spectrograms provide a natural representation of the spectral content of an audio

signal, with greater resolution in lower frequencies and reduced resolution in higher frequencies. They are commonly used as input features for machine learning models in tasks such as speech and music processing.




### B. Logistic Regression

Logistic Regression is a widely used statistical technique for binary classification tasks. In the context of music genre classification, it can be adapted to handle multiple classes using techniques such as one-vs-rest or multinomial logistic regression. Logistic Regression models the probability that a given input belongs to each class using a logistic function, making it suitable for probabilistic classification tasks.

### C. K-nearest Neighbors (KNN)

K-nearest Neighbors is a simple yet effective non-parametric algorithm used for classification tasks. In KNN, the class of a test sample is determined by a majority vote among its k nearest neighbors in the training dataset, where the distance metric is typically Euclidean distance. KNN is known for its simplicity and intuitive concept, but it may suffer from high computational costs, especially with large datasets.

### D. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks have gained prominence in various fields, including computer vision and audio processing, due to their ability to automatically learn hierarchical feature representations from raw data. In the context of music genre classification, CNNs can learn to extract relevant features from spectrogram representations of audio signals, capturing both local and global patterns in the data. CNN architectures typically consist of convolutional layers followed by pooling layers and fully connected layers, allowing them to model complex relationships in the data.

### E. Decision Trees

Decision Trees are a popular class of algorithms that recursively partition the feature space into regions, making decisions based on the values of input features. While decision trees are interpretable and easy to understand, they may suffer from overfitting and lack of generalization. However, ensemble methods like Random Forests and Gradient Boosted Trees can mitigate these issues and often yield better performance.

### F. RCNN (Recurrent Convolutional Neural Network)

RCNN (Recurrent Convolutional Neural Network) combines the strengths of both recurrent neural networks (RNNs) and CNNs, allowing it to model sequential and spatial dependencies in the data simultaneously. RCNN has shown promising results in various sequential classification tasks, including music genre classification, by capturing both temporal and spectral features of audio signals.

Overall, each algorithm has its advantages and limitations, and the choice of algorithm depends on factors such as dataset size, feature representation, and computational resources. In this project, we aim to compare the performance of Logistic Regression, KNN, CNN, Decision Trees, and RCNN on the task of music genre classification using the GTZAN dataset.

## V. METHODOLOGY

### A. K-Nearest Neighbors (KNN)

We preprocess the data by splitting it into training and testing sets and perform feature scaling using standardization. KNN works by finding the k-nearest neighbors of a given data point based on a distance metric (e.g., Euclidean distance) in the feature space. The class label for the data point is then determined by majority voting among its k-nearest neighbors. We then train a KNN classifier and evaluate its performance in terms of accuracy, classification report, and confusion matrix analysis.

### B. Logistic Regression

We preprocess the data by standardizing the features and splitting the dataset into training and testing sets. Logistic regression is a linear model that predicts the probability of a binary outcome (or multiple outcomes with multinomial logistic regression) based on input features. It works by modeling the relationship between the categorical dependent variable and one or more independent variables using the logistic function. We then train a logistic regression model using scikit-learn and evaluate its performance in terms of accuracy, classification report, and confusion matrix analysis.

### C. Convolutional Neural Networks (CNNs)

We preprocess the data by encoding the labels and splitting the dataset into train and test sets. CNNs are deep learning models particularly well-suited for image processing tasks but can also be applied to sequential data like audio. CNNs operate by passing input data through a series of convolutional and pooling layers, which learn hierarchical representations of features at different levels of abstraction. The extracted features are then flattened and passed through fully connected layers for classification.

*1) Model Architecture:* The CNN model architecture is built using the Sequential API from TensorFlow and Keras, comprising the following layers:

- Convolutional layers with 1D filters of sizes 64, 128, 256, and 512, each followed by ReLU activation functions.
- Batch normalization layers after each convolutional layer to stabilize and accelerate the training process.
- Max pooling layers with pool size of 2 after each convolutional layer to downsample the feature maps.
- Flatten layer to convert the 3D output of the convolutional layers into 1D feature vectors.
- Fully connected dense layers with 512 and 256 units, each followed by ReLU activation functions and dropout layers with dropout rate of 0.5 to prevent overfitting.

- Output dense layer with softmax activation function, having units equal to the number of classes in the dataset, for classification.

*2) Model Compilation and Training:* The model is compiled using the Adam optimizer and sparse categorical crossentropy loss function. It is trained on the training dataset for 12 epochs with a batch size of 32. Additionally, a validation split of 0.1 is used for monitoring the model's performance during training.

### *D. Decision Tree Classifier*

We preprocess the data by splitting it into training and testing sets and perform feature scaling using standardization. Decision trees are non-parametric supervised learning models used for classification and regression tasks. The algorithm recursively splits the data based on features to create a tree-like structure, where each internal node represents a feature and each leaf node represents a class label or regression value. We then train a Decision Tree Classifier and evaluate its performance in terms of accuracy, classification report, confusion matrix analysis, and classification report.

### *E. CNN MOdel 2*

We first load grayscale images from the directory created before to store pre-processed melspectrogram, resizes them to a fixed size, and normalizes pixel values. It encodes class labels and splits the data into training and testing sets. This preprocessing prepares the data for training the CNN model. Model Architecture:

- Convolutional Layers:
  - Conv2D(16, kernel_size=(3, 3), activation='relu', padding='same')
  - MaxPooling2D(pool_size=(2, 2))
  - Conv2D(32, kernel_size=(3, 3), activation='relu', padding='same')
  - MaxPooling2D(pool_size=(2, 2))
  - Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same')
  - MaxPooling2D(pool_size=(2, 2))
  - Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same')
  - MaxPooling2D(pool_size=(2, 2))
  - Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same')
  - MaxPooling2D(pool_size=(2, 2))
- Flatten Layer
- Dense Layers:
  - Dense(128, activation='relu')
  - Dropout(0.2)
  - Dense(64, activation='relu')
  - Dropout(0.2)
  - Dense(32, activation='relu')
  - Dropout(0.2)
  - Dense(num_classes, activation='softmax')

### *F. RCNN (Recurrent Convolutional Neural Network)*

We first load grayscale images from the directory created before to store pre-processed melspectrogram, resizes them to a fixed size, and normalizes pixel values. It encodes class labels and splits the data into training and testing sets. This preprocessing prepares the data for training the RCNN model.

Model Architecture:

- Input Layer: The input layer accepts data in the shape specified by `input_shape`.
- Convolutional Layers: Successive convolutional layers (16, 32, 64, 128 filters) extract spatial features with ReLU activation.
- MaxPooling Layers: After each convolution, MaxPooling layers (2x2) downsample feature maps, reducing spatial dimensions.
- Flatten Layer: Converts the output of convolutions into a one-dimensional vector.
- Reshape Layer: Rearranges the flattened output into a 3D tensor for recurrent layers.
- Recurrent Layers (GRU): Two GRU layers (128, 64 units) learn sequential patterns. First returns sequences, second returns final output.
- Dense Layers: Fully connected layers with ReLU activation perform classification.
- Dropout Layers: Prevent overfitting by randomly deactivating neurons (20
- Output Layer: Final dense layer with softmax activation predicts class probabilities.
- Compilation: Model compiled with Adam optimizer, sparse categorical cross-entropy loss, and accuracy metric.

## VI. RESULTS

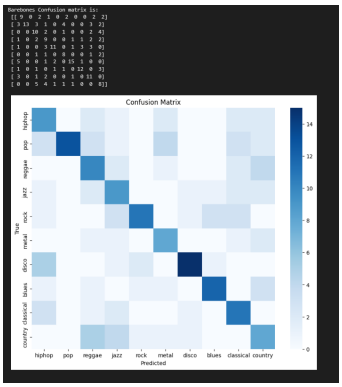
### *A. Decision Tree Classifier Performance*

*1) Accuracy:* Test Accuracy - 53%

*2) Classification Report:* A classification report provides detailed metrics such as precision, recall, and F1-score for each class in the Decision Tree Classifier.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| blues | 0.39 | 0.50 | 0.44 | 18 |
| classical | 1.00 | 0.45 | 0.62 | 29 |
| country | 0.40 | 0.53 | 0.45 | 19 |
| disco | 0.38 | 0.50 | 0.43 | 18 |
| hiphop | 0.73 | 0.50 | 0.59 | 22 |
| jazz | 0.47 | 0.62 | 0.53 | 13 |
| metal | 0.79 | 0.62 | 0.70 | 24 |
| pop | 0.71 | 0.63 | 0.67 | 19 |
| reggae | 0.46 | 0.61 | 0.52 | 18 |
| rock | 0.35 | 0.40 | 0.37 | 20 |
| accuracy | | | 0.53 | 200 |
| macro avg | 0.57 | 0.54 | 0.53 | 200 |
| weighted avg | 0.60 | 0.53 | 0.54 | 200 |

*3) Confusion Matrix:* A confusion matrix analysis is conducted to assess the Decision Tree Classifier's performance across different music genres.



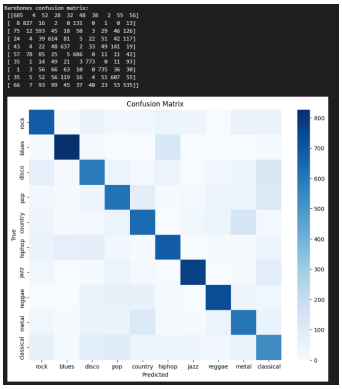

### B. Logistic Regression Performance

*1) Accuracy:* Test Accuracy - 66%

*2) Classification Report:* A classification report provides detailed metrics such as precision, recall, and F1-score for each class in the logistic regression model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| blues | 0.67 | 0.69 | 0.68 | 1000 |
| classical | 0.88 | 0.83 | 0.85 | 998 |
| country | 0.58 | 0.59 | 0.59 | 997 |
| disco | 0.59 | 0.61 | 0.60 | 999 |
| hiphop | 0.62 | 0.64 | 0.63 | 998 |
| jazz | 0.69 | 0.69 | 0.69 | 1000 |
| metal | 0.85 | 0.77 | 0.81 | 1000 |
| pop | 0.77 | 0.73 | 0.75 | 1000 |
| reggae | 0.61 | 0.61 | 0.61 | 1000 |
| rock | 0.49 | 0.54 | 0.51 | 998 |
| accuracy | | | 0.67 | 9990 |
| macro avg | 0.68 | 0.67 | 0.67 | 9990 |
| weighted avg | 0.68 | 0.67 | 0.67 | 9990 |

*3) Confusion Matrix:* A confusion matrix analysis is conducted to assess the logistic regression model's performance across different music genres.



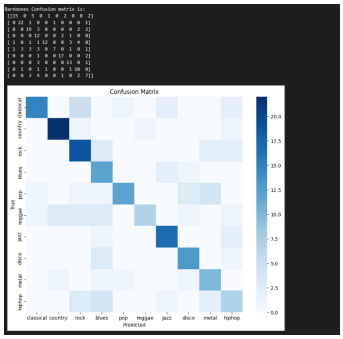

### C. KNN Performance

*1) Accuracy:* Test accuracy - 67%

*2) Classification Report:* A classification report provides detailed metrics such as precision, recall, and F1-score for each class in the KNN model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| blues | 0.88 | 0.60 | 0.71 | 25 |
| classical | 0.85 | 0.88 | 0.86 | 25 |
| country | 0.59 | 0.73 | 0.66 | 26 |
| disco | 0.43 | 0.80 | 0.56 | 15 |
| hiphop | 0.86 | 0.55 | 0.67 | 22 |
| jazz | 0.88 | 0.37 | 0.52 | 19 |
| metal | 0.77 | 0.85 | 0.81 | 20 |
| pop | 0.68 | 0.76 | 0.72 | 17 |
| reggae | 0.56 | 0.71 | 0.62 | 14 |
| rock | 0.44 | 0.41 | 0.42 | 17 |
| accuracy | | | 0.67 | 200 |
| macro avg | 0.69 | 0.67 | 0.66 | 200 |
| weighted avg | 0.71 | 0.67 | 0.67 | 200 |

*3) Confusion Matrix:* A confusion matrix analysis is conducted to assess the KNN model's performance across different music genres.




### D. CNN Performance

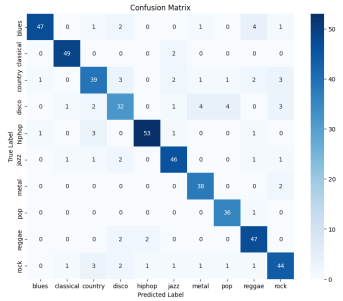
*1) Accuracy:* Test accuracy - 86%

*2) Classification Report:* A classification report provides detailed metrics such as precision, recall, and F1-score for each class in the CNN model.

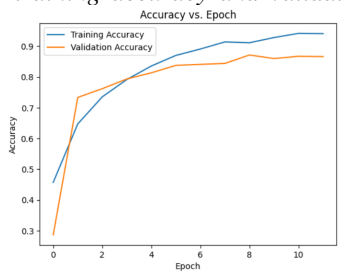
Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| blues | 0.96 | 0.84 | 0.90 | 56 |
| classical | 0.94 | 0.96 | 0.95 | 51 |
| country | 0.80 | 0.75 | 0.77 | 52 |
| disco | 0.74 | 0.68 | 0.71 | 47 |
| hiphop | 0.95 | 0.90 | 0.92 | 59 |
| jazz | 0.87 | 0.88 | 0.88 | 52 |
| metal | 0.84 | 0.95 | 0.89 | 40 |
| pop | 0.86 | 0.97 | 0.91 | 37 |
| reggae | 0.82 | 0.92 | 0.87 | 51 |
| rock | 0.81 | 0.80 | 0.81 | 55 |
| accuracy | | | 0.86 | 500 |
| macro avg | 0.86 | 0.87 | 0.86 | 500 |
| weighted avg | 0.86 | 0.86 | 0.86 | 500 |

*3) Confusion Matrix:* A confusion matrix analysis is conducted to assess the CNN model's performance across different music genres.




*4) Epoch vs training accuracy and validation accuracy:*



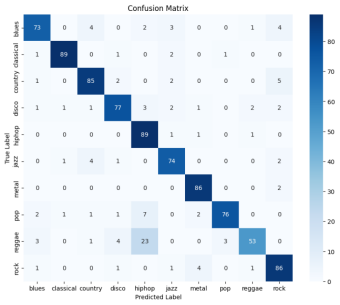

### E. CNN model 2 Performance

*1) Accuracy:* Test accuracy - 87.5%

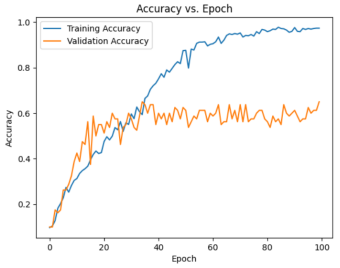
*2) Classification Report:* A classification report provides detailed metrics such as precision, recall, and F1-score for each class in the CNN model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| blues | 0.60 | 0.43 | 0.50 | 21 |
| classical | 0.62 | 0.83 | 0.71 | 12 |
| country | 0.45 | 0.58 | 0.51 | 24 |
| disco | 0.75 | 0.27 | 0.40 | 22 |
| hiphop | 0.54 | 0.87 | 0.67 | 15 |
| jazz | 0.65 | 0.74 | 0.69 | 27 |
| metal | 0.83 | 0.83 | 0.83 | 18 |
| pop | 0.50 | 0.37 | 0.42 | 19 |
| reggae | 0.67 | 0.64 | 0.65 | 22 |
| rock | 0.27 | 0.30 | 0.29 | 20 |
| accuracy | | | 0.57 | 200 |
| macro avg | 0.59 | 0.59 | 0.57 | 200 |
| weighted avg | 0.59 | 0.57 | 0.56 | 200 |

*3) Confusion Matrix:* A confusion matrix analysis is conducted to assess the CNN model's performance across different music genres.




*4) Epoch vs training accuracy and validation accuracy:*



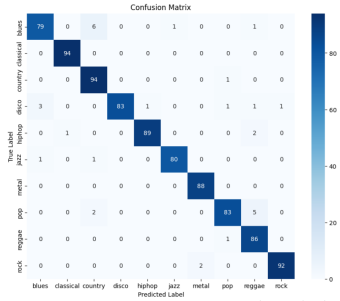

## *F. RCNN Performance*

*1) Accuracy:* Test accuracy - 89%

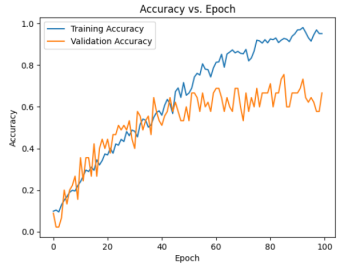
*2) Classification Report:* A classification report provides detailed metrics such as precision, recall, and F1-score for each class in the CNN model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| blues | 0.89 | 0.75 | 0.81 | 87 |
| classical | 0.99 | 1.00 | 0.99 | 94 |
| country | 0.84 | 0.91 | 0.87 | 95 |
| disco | 0.77 | 0.81 | 0.79 | 90 |
| hiphop | 0.89 | 0.90 | 0.90 | 92 |
| jazz | 0.94 | 0.99 | 0.96 | 82 |
| metal | 0.88 | 0.95 | 0.92 | 88 |
| pop | 0.86 | 0.80 | 0.83 | 90 |
| reggae | 0.93 | 0.85 | 0.89 | 87 |
| rock | 0.82 | 0.84 | 0.83 | 94 |
| accuracy | | | 0.88 | 899 |
| macro avg | 0.88 | 0.88 | 0.88 | 899 |
| weighted avg | 0.88 | 0.88 | 0.88 | 899 |

*3) Confusion Matrix:* A confusion matrix analysis is conducted to assess the CNN model's performance across different music genres.




*4) Epoch vs training accuracy and validation accuracy:*




## VII. CONCLUSION

Our study demonstrates the effectiveness of KNN, logistic regression, CNNs, and decision trees in music genre classification tasks, providing insights into their performance and potential applications in real-world scenarios.

## REFERENCES


[1] Cover, T., Hart, P. (1967). Nearest neighbor pattern classification.
[2] Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning.
[3] Krizhevsky, A., Sutskever, I., Hinton, G. (2012). ImageNet classification with deep convolutional neural networks.
[4] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees.
[5] Quinlan, J. R. (1986). Induction of decision trees.
[6] Kozakowski, P., Michalak, B. 1D convolution model
[7] Ghosh, P., Mahapatra, S., Jana, S., Jha, R. K. RCNN model architecture.
[8] Young Park, from Geek Culture for pre-processing in melspectrogram.