Machine Learning Assignment

PROJECT REPORT

TEAM ID: 31

Music Genre Classification from Audio Signals

Name	SRN
Chinthan K	PES2UG23CS155
Dhruv Hegde	PES2UG23CS172

Problem Statement

The increasing volume of digital music necessitates an efficient, automated method for organization and retrieval. Manual categorization is a time-consuming and subjective process. This project aims to address the challenge of accurately classifying music recordings by genre using machine learning. The goal is to develop and evaluate a robust model capable of classifying music into one of 10 popular genres by extracting and analyzing key audio features. This project will also compare different machine learning models, such as Random Forests, Support Vector Machines (SVMs), and simple Convolutional Neural Networks (CNNs), to determine the most effective classifier.

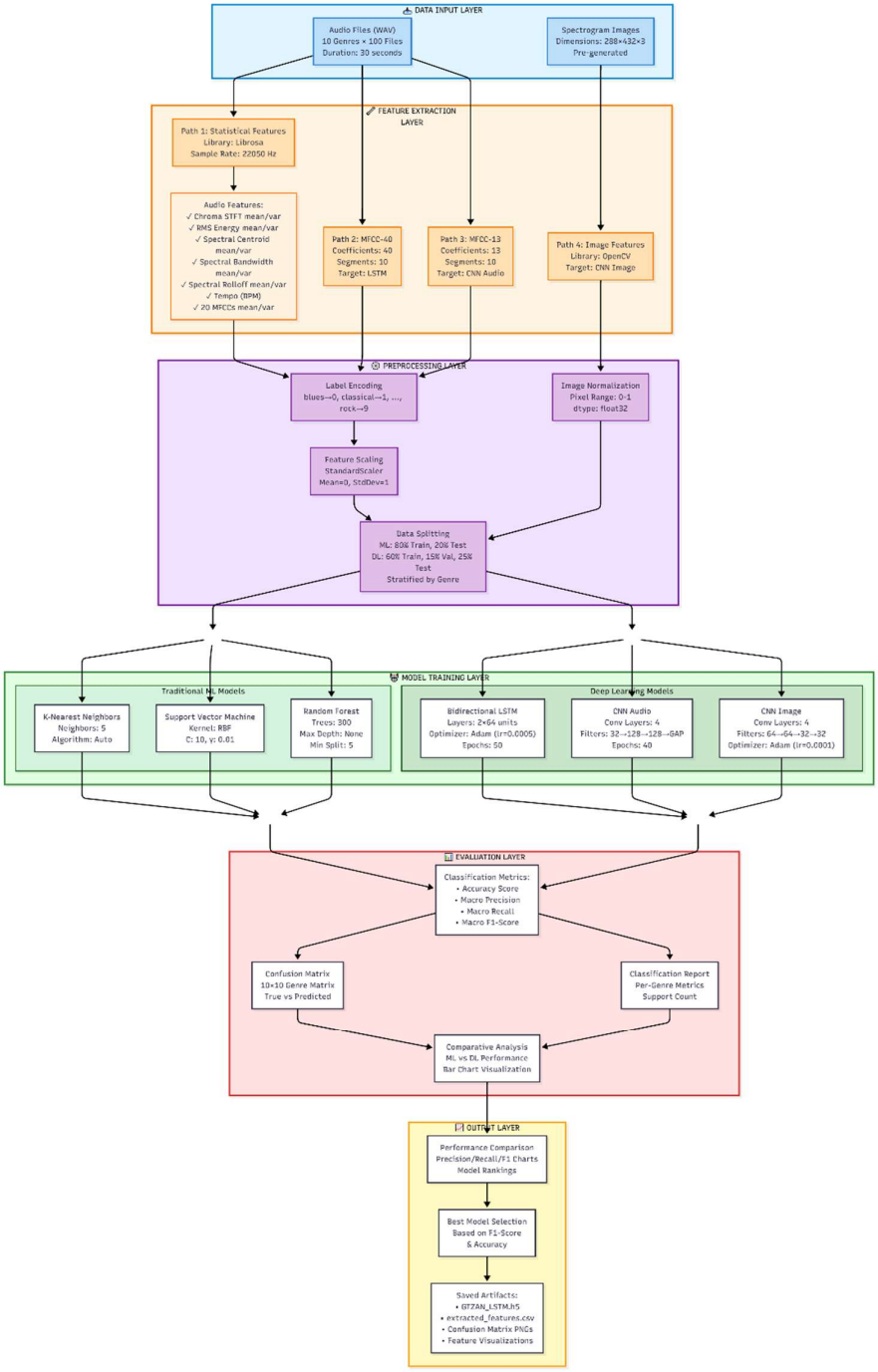
Objective / Aim

The objective of this project is to build and evaluate a model that can accurately classify music clips into their respective genres based on audio features. The project aims to achieve high performance, as measured by key evaluation metrics, while demonstrating a practical application of audio signal processing and deep learning. Specifically, the project will compare the effectiveness of different feature sets (e.g., MFCCs vs. spectrograms) and model architectures to identify the optimal model for this classification task. This comparison will provide insights into the strengths and weaknesses of each approach for music genre categorization.

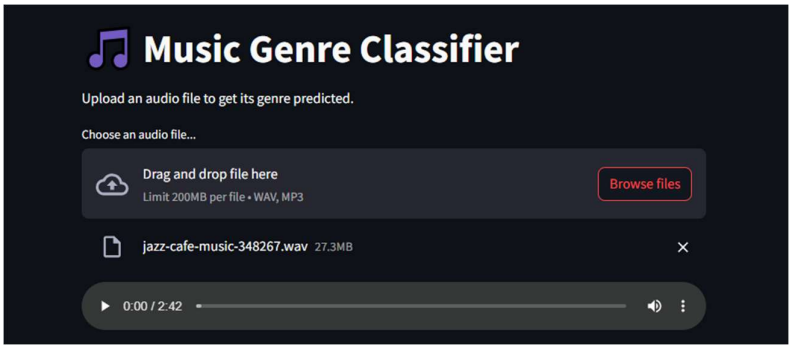
Dataset Details

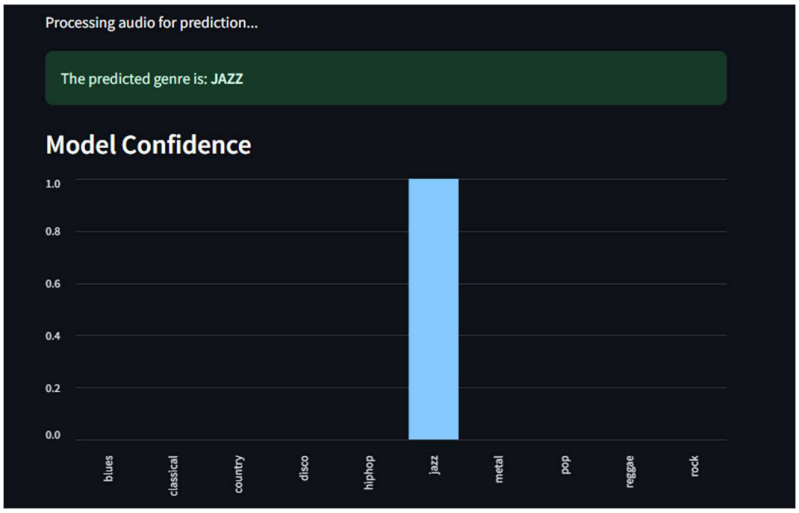
- **Source:** Kaggle (GTZAN Dataset)
- **Size:** 1,000 audio files, each 30 seconds long, distributed across 10 genres.
- **Key Features:** Mel-frequency cepstral coefficients (MFCCs), Spectrograms
- **Target Variable:** Music Genre (e.g., blues, rock, classical).

Architecture Diagram



USER INTERFACE:





- The application uses Streamlit for the web interface
- Backend powered by TensorFlow for deep learning model inference
- Uses librosa for audio feature extraction including: Chroma STFT features RMS energy Spectral features (centroid, bandwidth, rolloff) MFCCs (Mel-frequency cepstral coefficients)
- Implements robust error handling for audio processing
- Supports both MP3 and WAV file formats
- Uses StandardScaler for feature normalization
- Employs a CNN model architecture

Methodology

Traditional Machine Learning Models Implementation

This section details the implementation approach for the traditional machine learning models used for music genre classification based on the extracted audio features.

Approach:

Data Preparation:

* The extracted features DataFrame is used as the input data.
* The features (X) are separated from the labels (y).
* The genre labels (y) are encoded numerically using LabelEncoder.
* The feature data (X) is scaled using StandardScaler to ensure that all features contribute equally to the model training.
* The data is split into training and testing sets (X_train, X_test, y_train, y_test) using train_test_split, with stratification to maintain the genre distribution in both sets.

Model Implementation:

* **K-Nearest Neighbors (KNN):**
  * A KNeighborsClassifier is initialized with n_neighbors=5.
  * The model is trained on the scaled training data.
  * Predictions are made on the scaled test data .
* Support Vector Machine (SVM):
  * A SVC model is initialized with a radial basis function kernel , a regularization parameter C=10, and a kernel coefficient gamma=0.01..
  * The model is trained on the scaled training data.
  * Predictions are made on the scaled test data.
* Random Forest:
  * A Random Forest Classifier is initialized with specific parameters (n_estimators=300, max_depth=None, min_samples_leaf=1, min_samples_split=5, random_state=42). These parameters are noted as best_rf_params, suggesting they were the result of hyperparameter tuning (as shown in a later cell).
  * The model is trained on the scaled training data.

* Predictions are made on the scaled test data.

LSTM Deep Learning

Model Approach:

Data Preprocessing:

- Audio files are loaded and segmented.
- Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from each segment.
- The data is then split into training, validation, and testing sets.

Model Architecture:

- The model is a Sequential Keras model.
- It uses two Bidirectional LSTM layers to capture temporal dependencies.
- A Dense layer with ReLU activation is followed by a final Dense layer with softmax for genre classification.

Training:

- The model is compiled with the Adam optimizer and sparse categorical crossentropy loss.
- It is trained for 50 epochs with a batch size of 64.

CNN Deep Learning

Model(image) Approach:

Data Loading and Preprocessing:

- Images are loaded from the /kaggle/input/musicdata/Data/images_original directory.
- Labels are extracted and numerically encoded using LabelEncoder.
- The data is split into training, validation, and testing sets.
- Pixel values are normalized by scaling them to a range of 0 to 1.

Model Architecture:

- The model is a Sequential Keras model.
- It uses multiple Conv2D and MaxPooling2D layers to extract and downsample image features.
- A Dropout layer is included to prevent overfitting.
- The output is flattened and fed into Dense layers.

- The final Dense layer with softmax activation provides the probability of each of the 10 genres.

Training:

- The model is compiled with the Adam optimizer and sparse_categorical_crossentropy loss.
- It is trained for 50 epochs with a batch size of 32.

CNN Deep Learning Model (MFCCs)(Music) Approach:

Data Preprocessing:

- MFCCs are extracted from audio files.
- The MFCC data is reshaped to be compatible with a CNN, and labels are one-hot encoded.
- The data is then split into training, validation, and testing sets.

Model Architecture:

- The model is a Sequential Keras model.
- It uses multiple Conv2D and MaxPooling2D layers to learn features from the MFCC data.
- Dropout layers are included for regularization.
- A GlobalAveragePooling2D layer reduces spatial dimensions.
- Dense layers process the output before the final softmax activation for genre classification.

Training:

- The model is compiled using the Adam optimizer and binary_crossentropy loss.
- It is trained for 40 epochs with a batch size of 32.

Results & Evaluation

MODEL	PRECISION	RECALL	F1-SCORE	OVERALL ACCURACY
KNN	0.592	0.575	0.572	0.575
SVM	0.721	0.715	0.715	0.715
RANDOM FOREST	0.707	0.705	0.702	0.705
LSTM	0.77	0.76	0.76	0.76
CNN	0.89	0.89	0.89	0.89

- **KNN**: This model had the lowest overall performance, with an accuracy of 57.5%. Its precision, recall, and F1-scores were also low. The model performed better on genres like 'classical' and 'metal' but struggled with 'country', 'disco', 'hiphop', and 'reggae'. A confusion matrix would highlight these specific misclassifications.
- **SVM**: The SVM model outperformed the other traditional models, achieving the highest accuracy at 71.5%. Its precision, recall, and F1-scores were generally better and more consistent across genres. While it showed improvement for most categories, it still had lower F1-scores for genres such as 'disco' and 'reggae', though its confusion matrix would show fewer misclassifications than the KNN model.
- **Random Forest**: This model had an accuracy of 70.5%, performing slightly below the SVM but still significantly better than the KNN model. The precision, recall, and F1-scores were comparable to the SVM. It performed particularly well on 'blues', 'classical', 'metal', and 'pop', while also having lower scores for 'disco' and 'reggae'. A feature importance plot would reveal which features were most influential in its predictions.
- **LSTM Model**: This model achieved an overall accuracy of approximately **76%**. It performed well on genres like 'classical', 'jazz', 'metal', and 'rock', with good scores for precision, recall, and F1. However, its performance was weaker for genres such as 'hiphop', 'reggae', and 'country'. The confusion matrix for the LSTM model showed noticeable misclassifications between genres.
- **CNN Model**: The CNN model demonstrated significantly superior performance, achieving a higher overall accuracy of approximately **88.7%**. Its classification report and confusion matrix indicated superior performance across most genres compared to the LSTM. The CNN model had high precision, recall, and F1-scores for nearly all genres, with especially strong results for 'classical', 'disco', 'jazz', and 'metal'. Its confusion matrix had a much stronger diagonal, confirming more accurate classifications and a better ability to distinguish between genres.

Conclusion

The results of this project demonstrate that a Convolutional Neural Network (CNN) trained on Mel-Frequency Cepstral Coefficients (MFCCs) is the most effective approach among the evaluated models for music genre classification on the GTZAN dataset. The CNN's ability to learn spatial hierarchies in the MFCC data proved to be highly successful in distinguishing between different music genres. Future work could explore more advanced deep learning architectures, data augmentation techniques, or the fusion of multiple audio features to potentially further improve classification performance.

Through this project, we've gained valuable insights into applying various machine learning and deep learning techniques for audio classification. We've learned the importance of appropriate feature engineering, demonstrating how MFCCs effectively capture relevant audio characteristics. The comparison of traditional and deep learning models highlighted the power of deep architectures, particularly CNNs, in handling complex sequential data like audio features. We also saw how different data representations (raw features vs. image representations) can significantly impact model performance. Furthermore, the process of data splitting, scaling, model building, training, and evaluation provided practical experience in a typical machine learning workflow.