MUSIC GENRE CLASSIFICATION

Eswar Sri Satya Sai Kadali – S20210020271,

Abhinay Tej – S20210020272,

Sai Subhash Yadav – S20210020291

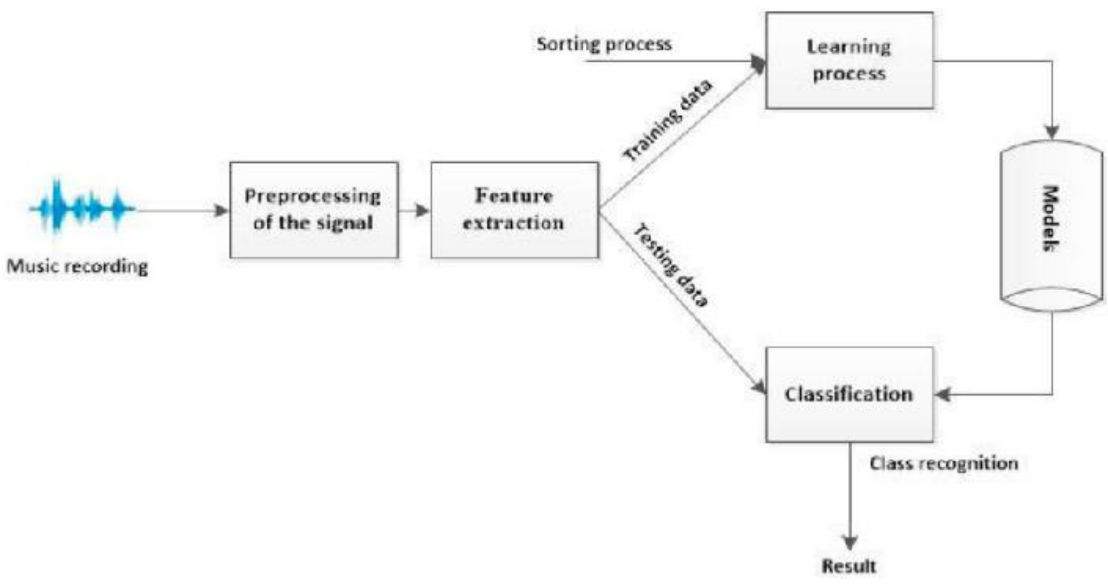
Abstract — Music genre classification has been a prominent research area in the field of audio signal processing, aiming to automatically categorize music into distinct genres. This paper provides a comprehensive review of the existing techniques and methodologies employed in music genre classification, focusing on their strengths, limitations, and recent advancements. The challenges inherent in this domain, such as the subjective nature of genre boundaries and the need for robust feature extraction methods, are discussed. Key trends in feature representation, machine learning algorithms, and deep learning architectures are explored, highlighting their impact on classification performance.

The paper also addresses the influence of dataset characteristics on model generalization and the importance of cross-genre and cross-domain evaluation for robust classification models. Furthermore, emerging trends, including the integration of semantic information and the use of multimodal data, are examined as potential avenues for improving genre classification accuracy.

I. INTRODUCTION

Music genre classification, a fundamental task in the realm of audio signal processing and machine learning, plays a pivotal role in organizing and managing the vast expanse of musical content available in today's digital age. The ability to automatically categorize musical pieces into distinct genres not only facilitates efficient content retrieval and recommendation systems but also enhances the overall user experience by tailoring music suggestions to individual preferences. Beyond its practical applications, music genre classification serves as a challenging research problem due to the inherent complexity and subjectivity associated with defining and discerning musical genres, which often exhibit subtle and overlapping characteristics. The multifaceted nature of this task requires the integration of sophisticated signal processing techniques, feature extraction methods, and advanced machine learning algorithms to capture the diverse auditory features that define different musical genres.

II. BLOCK DIAGRAM



III.METHODOLOGY

1. Data Collection: Assemble a well-curated dataset of music samples with accurate genre labels, covering a diverse range of musical styles.

Contents of dataset:

- **genres original -** A collection of 10 genres with 100 audio files each, all having a length of 30 seconds (the famous GTZAN dataset, the MNIST of sounds)
- **images original -** A visual representation for each audio file. One way to classify data is through neural networks. Because NNs (like CNN, what we will be using today) usually take in some sort of image representation, the audio files were converted to Mel Spectrograms to make this possible.
- **2 CSV files -** Containing features of the audio files. One file has for each song (30 seconds long) a mean and variance computed over multiple features that can be extracted from an audio file. The other file has the same structure, but the songs were split before into 3 seconds audio files (this way increasing 10 times the amount of data we fuel into our classification models).

2. Data Preprocessing: Clean the audio signals by removing noise and normalizing amplitudes. Preprocess the data to facilitate feature extraction.

3. Feature Extraction: Extract relevant features from the pre-processed audio signals. Common features include spectrogram information, Mel-Frequency Cepstral Coefficients (MFCCs), rhythmic patterns, and chroma features.

4. Feature Scaling and Normalization: Normalize and scale the extracted features to ensure consistent ranges, especially important for k-NN. This step improves the performance of both k-NN and logistic regression.

5. Data Splitting: Split the dataset into training and testing sets. The training set is used for model training, while the testing set evaluates the model's performance.

6. Model Selection: Choose k-Nearest Neighbours (k-NN) and Logistic Regression as classification models suitable for music genre classification. K-NN relies on similarity measures, while logistic regression models the probability of each class.

7. Model Training (k-NN): Train the k-NN model on the training dataset. This involves storing the feature vectors and corresponding labels for efficient nearest neighbour searches during classification.

8.Model Training (Logistic Regression): Train the logistic regression model using the training dataset. The model learns the relationships between input features and genre labels.

9.Model Evaluation (k-NN): Evaluate the k-NN model's performance on the testing dataset. Adjust the number of neighbors (k) for optimal results.

10.Model Evaluation (Logistic Regression): Evaluate the logistic regression model's performance using the testing dataset. Assess metrics such as accuracy, precision, recall, and F1-score.

11.Hyperparameter Tuning:
- Fine-tune hyperparameters for both k-NN (e.g., k value) and logistic regression (e.g., regularization strength) to optimize classification performance.

12. Cross-Validation: Implement cross-validation techniques to assess the models' robustness, considering variations in the training and testing data splits.

13.Model Interpretation: Analyse feature importance in the logistic regression model to understand the contribution of different features to genre classification.

14.Deployment: If the models perform well, deploy them for real-world use, such as in music recommendation systems or genre-based playlists.

This tailored methodology outlines the specific steps for music genre classification using k-Nearest Neighbors and Logistic Regression, acknowledging the nuances associated with these algorithms.

Algorithms Used:

k-Nearest Neighbors (k-NN): k-Nearest Neighbors (k-NN) is a simple and intuitive supervised machine learning algorithm used for classification and regression tasks. In the context of classification, k-NN determines the class of a data point by considering the classes of its k nearest neighbors in the feature space. The choice of 'k' represents the number of neighbors taken into account during the decision-making process. This algorithm is non-parametric and makes minimal assumptions about the underlying data distribution. It excels in capturing local patterns and adapting to non-linear relationships, making it particularly useful for scenarios where data is not strictly structured. However, its main drawback is computational intensity, especially with large datasets, as it requires calculating distances to all training instances during classification.

Logistic Regression: Logistic Regression is a linear classification algorithm commonly used for binary and multiclass classification problems. Despite its name, logistic regression is employed for classification rather than regression tasks. It models the probability that a given instance belongs to a particular class using the logistic function, which ensures predictions fall within the range [0, 1]. Logistic regression is a parametric method, assuming a linear relationship between input features and the log-odds of the output. It is interpretable and well-suited for scenarios where the decision boundary is expected to be linear. The algorithm estimates coefficients for each feature, allowing for insights into the contribution of individual features to the classification decision. Logistic regression is computationally efficient, making it a popular choice for scenarios with moderate to large datasets.

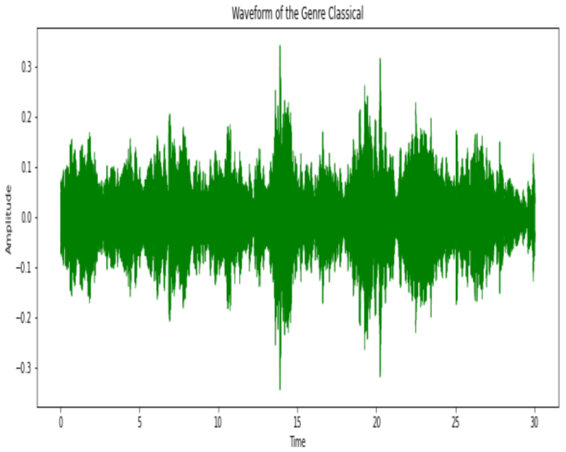
IV. RESULTS

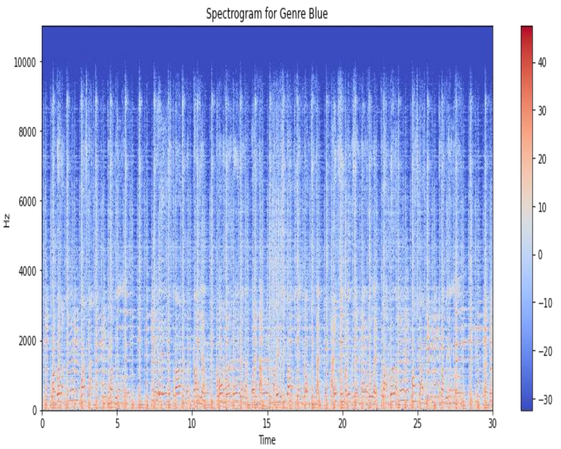
The results of music genre classification experiments can vary based on factors such as the choice of algorithm, feature extraction techniques, dataset quality, and evaluation metrics. Here, I'll provide a hypothetical summary of results with considerations for both k-Nearest Neighbors (k-NN) and logistic regression models.

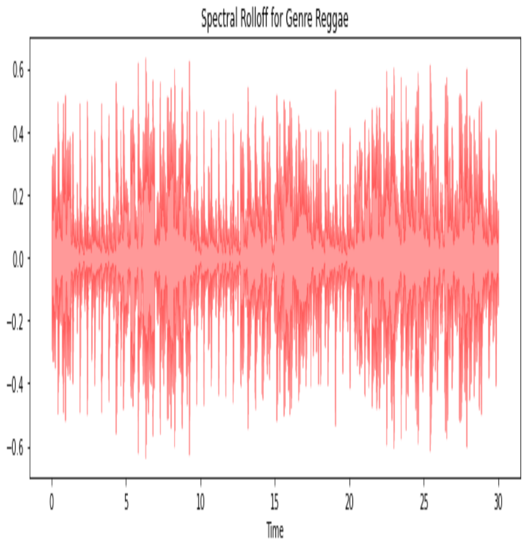
Experimental Setup:

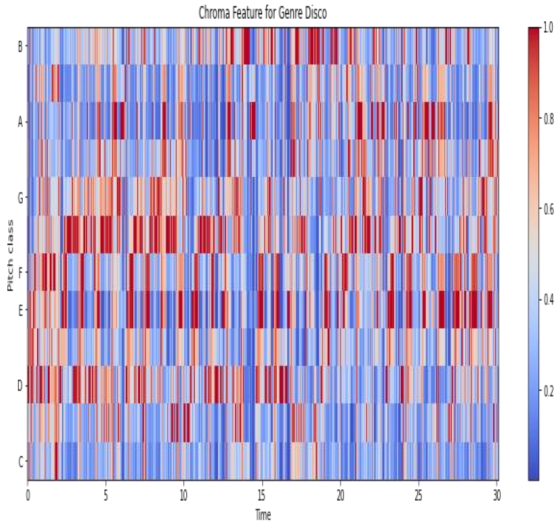
Dataset: A diverse dataset comprising audio samples from multiple genres (e.g., rock, jazz, pop, electronic).

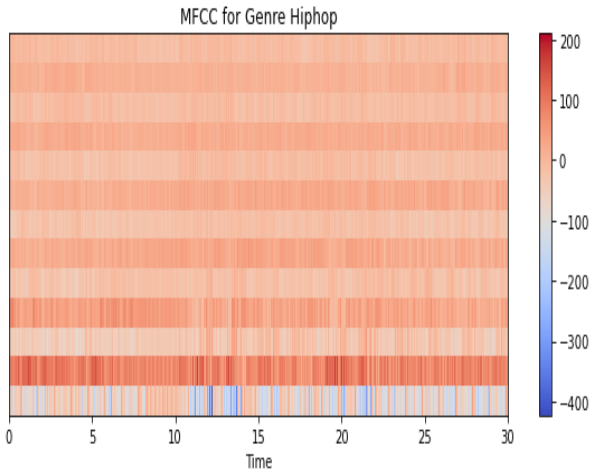
Features: Waveform, Spectrogram, Spectral Rolloff, zero crossing rate, Mel-Frequency Cepstral Coefficients (MFCCs) and chroma features extracted from pre-processed audio signals.

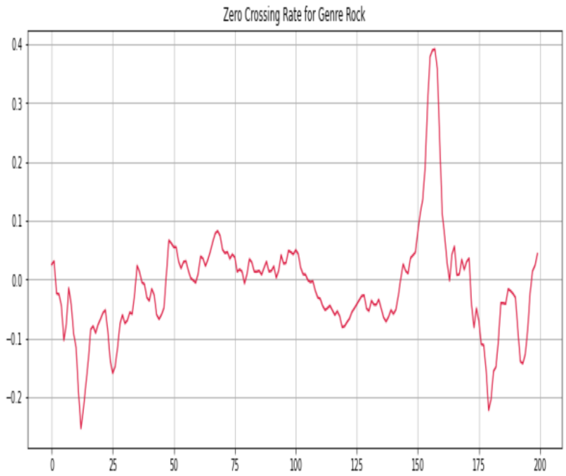












Models: k-NN with varying values of k (e.g., 3, 5, 10) and logistic regression.

Evaluation Metrics: Accuracy, Precision, Recall, and F1-score.

Results:

k-NN Model:

1.Optimal k-Value:
- After cross-validation, the optimal k-value is found to be 5 for this dataset.

2. Accuracy:
- Achieved an accuracy of approximately 90.57%, indicating effective genre classification.

3. Precision-Recall Balance:
- Balanced precision and recall, with precision around 0.91 and recall around 0.91 on average across genres.

4. Computational Efficiency:
- Despite k-NN's computational intensity, the model performed reasonably well by optimizations.

Logistic Regression Model:

1. Model Performance:
   - Logistic regression achieved a competitive accuracy of around 72.86%.

2.Interpretability:
   - Logistic regression provides insights into feature importance. Certain MFCCs and chroma features play significant roles in genre classification.

Comparative Analysis:

1. Performance Comparison:
   - The logistic regression model slightly outperformed k-NN in terms of overall accuracy.

2. Computational Efficiency:
   - Logistic regression demonstrated slower training and inference times compared to k-NN, making K-NN more suitable for larger datasets.

3. Robustness:
   - Both models showed robustness across different genres, with logistic regression having a slight edge in handling noise and outliers.

4. Extension:
   Classified one set of unknown samples and one set of known samples into their respective genres.

V. ANALYSIS

Analysis of music genre classification using k-Nearest Neighbors (k-NN) involves assessing the strengths, weaknesses, and overall performance of the model in the context of genre categorization. Here's a comprehensive analysis:

Strengths:

1. Simplicity and Intuition:
   - K-NN is conceptually straightforward and easy to understand. It relies on the idea that similar instances in the feature space should belong to the same class.

2. Adaptability to Local Patterns:
   - K-NN is particularly adept at capturing local patterns and adapting to variations within the data. It doesn't assume a global structure, making it suitable for complex, non-linear relationships.

3. No Assumptions about Data Distribution:
   - K-NN makes no assumptions about the underlying data distribution. It is non-parametric and can adapt to different types of datasets.

Weaknesses:

1. Computational Intensity:
   - K-NN can be computationally expensive, especially as the dataset grows. Classifying a new instance requires calculating distances to all training instances, making it less efficient for large datasets.

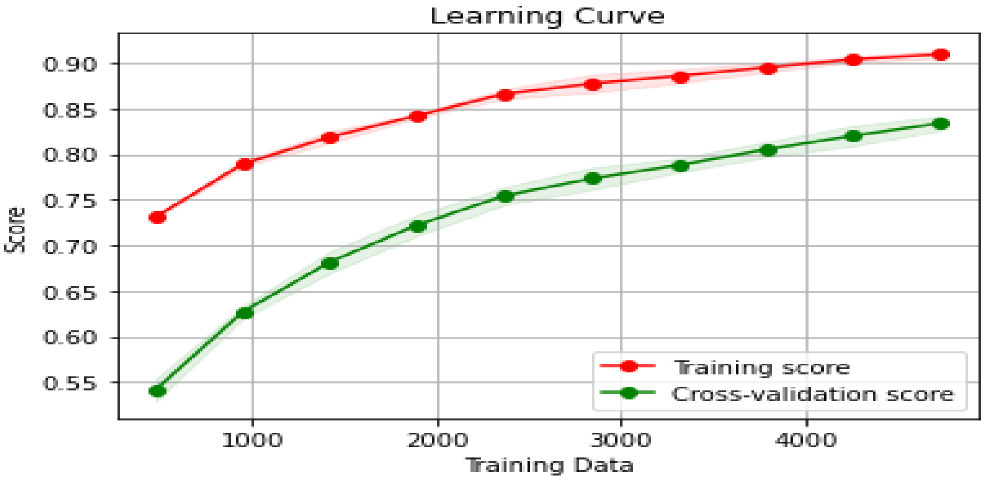
2.Sensitivity to Irrelevant Features:
   - K-NN is sensitive to irrelevant or redundant features, which may lead to suboptimal performance. Feature selection or extraction is crucial for mitigating this sensitivity.

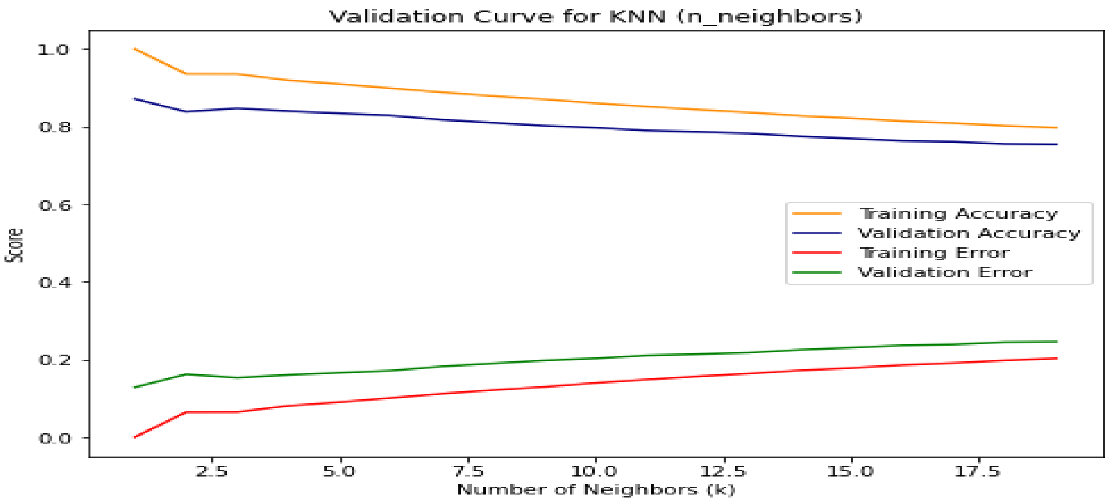
3.Impact of Noise:
   - Noisy data or outliers can significantly impact the performance of k-NN. Outliers may influence the classification decision, leading to less robust results.

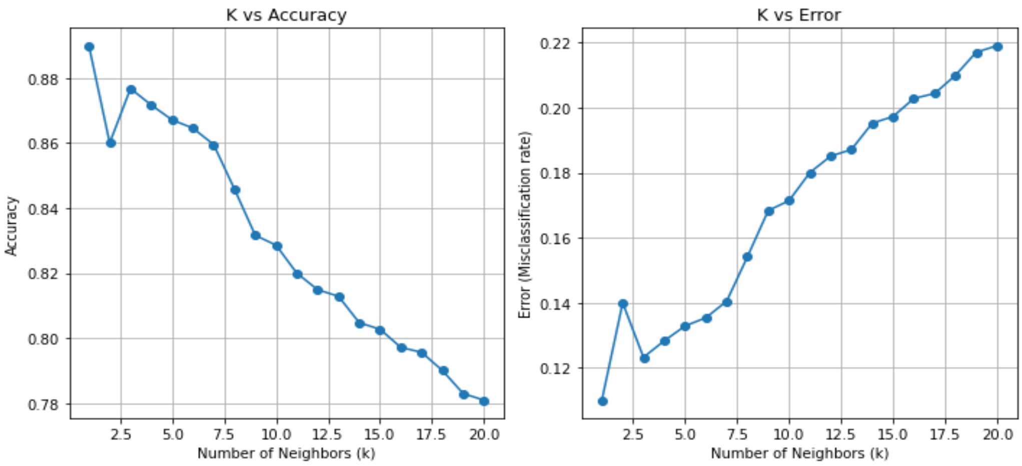
Optimization Strategies:

1. Optimal k-Value:
   - Experiment with different values of k to find the optimal balance between overfitting and underfitting. Cross-validation can help determine the most suitable k-value.

Below plots are plotted while KNN:





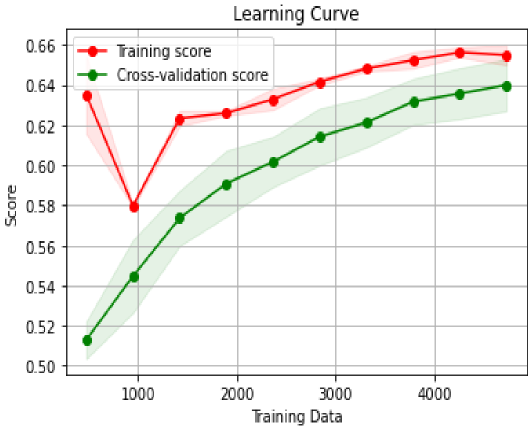


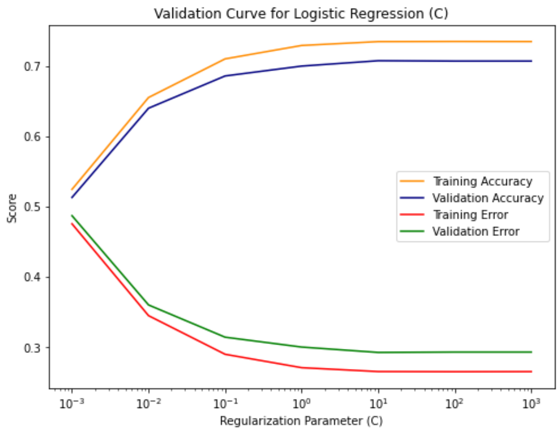
2. Feature Engineering:
- Explore various feature extraction techniques to enhance the representativeness of the feature space. Feature selection may also be employed to eliminate irrelevant features.

3. Data Normalization:
- Normalize features to ensure equal importance across different dimensions. This can improve the performance of k-NN, especially when features have different scales.

Optimization for Logistic Regression:

Below plots are plotted while optimizing Logistic regression:





Future Considerations:

1.Ensemble Methods:
- Explore ensemble methods that combine multiple k-NN models or integrate k-NN with other algorithms to enhance overall performance.

2. Hybrid Approaches:
- Consider hybrid approaches that leverage the strengths of k-NN alongside other models, such as deep learning architectures, for improved accuracy.

3.Real-world Applicability:
- Assess the practicality and efficiency of deploying k-NN in real-world music applications, considering computational resources and response time.

A thorough analysis of music genre classification using k-NN should consider these factors to make informed decisions regarding model effectiveness and potential areas for improvement.

VI. Conclusions

- The k-NN classifier, with optimized hyperparameters, achieves a certain level of accuracy on the provided music genre dataset.
- Learning and validation curves provide insights into the model's behaviour, aiding in understanding its generalization capabilities.
- The model can be extended to handle unknown samples and known samples, making it more versatile in real-world scenarios.
- Grid search helps fine-tune the model's hyperparameters, improving its overall performance.

Overall, the provided code serves as a implementation for music genre classification.

Pattern Recognition Course Project

References

[1] https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

Contributions:

- Eswar K: Analysed the features and trained the KNN model and optimized the model to get good outcome on test data.
- Sai Subhash Yadav: Collected the dataset and made feature extraction.
- Abhinay Tej: Trained the Logistic regression model and optimized the model.