Music Genre Classification

Deepak Bhatter Tushar Bhatt Prajjwal Dixit
Rhythm Patni Rahul Sharma Mayank Agarwal

April 21, 2024

Abstract: Explore music genre classification using machine learning to develop an accurate genre prediction system.
Given: A dataset of music audio files from diverse genres and the acoustic features extracted using Librosa library.
Outcome: A trained classification model to learn patterns, and correctly predict genre of a random music file.
Keywords: Music genre classification, Machine learning, Acoustic features, Dataset, Model training, Evaluation, Prediction.

Contents

1 Introduction

1.1 Background

The classification of music into genres is a task that has long fascinated researchers and practitioners in the field of musicology and music information retrieval. Traditionally, this classification was done manually by experts, a process that is not only time-consuming but also prone to inconsistencies due to the subjective nature of music genres. With the advent of digital music and the explosion of music data available online, automated genre classification has become an important tool in managing large music databases.

The concept of music genre is inherently complex because it is based on a combination of factors such as tempo, rhythm, instruments, performance style, and even cultural and historical contexts. Despite this complexity, researchers have approached the challenge of genre classification using various techniques in signal processing and machine learning.

1.2 Objective of this Project

The objective of this project is to develop and evaluate a series of machine learning models capable of accurately classifying music tracks into their respective genres based on their audio features. It aims to compare the effectiveness of various classification algorithms, including K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machines (SVM), AdaBoost, Logistic Regression, and Artificial Neural Networks (ANN), to determine which method provides the most reliable and accurate genre predictions.

1.3 Need & Real-Life Application

Music genre classification is integral to music streaming platforms like Spotify, Apple Music, and many more. These services rely on accurate genre classification to enhance user experience by offering personalized playlists and recommendations. Libraries, archives, and online music databases can utilize advanced genre classification systems to organize vast collections of music more efficiently. This aids in quick retrieval and easier management of music files, making it simpler for users to find tracks of interest.

2 Approaches Tried

The code has been structured into six sections for systematic presentation. As for techniques used, we have used six techniques namely: K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM), AdaBoost, Logistic Regression, and Artificial Neural Network (ANN).

2.1 K-Nearest Neighbour (KNN)

K-Nearest Neighbors (KNN) is a simple classification algorithm that determines the class label of a data point by considering the majority class among its k nearest neighbors in the feature space. It operates on the principle of similarity, where the class of an unknown sample is determined by the classes of its closest neighbors. KNN is non-parametric and does not require explicit model training, making it easy to understand and implement.

2.2 Decision Tree

Decision Trees are hierarchical structures where each internal node represents a decision based on a feature value, splitting the dataset into subsets. This recursive partitioning continues until leaf nodes are reached, providing the final classification. Nodes are determined by selecting features that best separate the data. This method provides simpler decision-making processes and is useful for both classification and regression tasks in machine learning.

2.3 Support Vector Machines (SVMs)

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It constructs a hyperplane in the feature space to effectively separate data points belonging to different classes while maximizing the margin between them. By identifying the optimal hyperplane, SVM aims to achieve the greatest possible separation between classes, making it robust and effective for various classification problems in machine learning.

2.4 AdaBoost

AdaBoost, short for Adaptive Boosting, is an ensemble learning method that iteratively trains weak learners on data, emphasizing misclassified samples in each iteration. By assigning weights to training instances based on their classification accuracy, AdaBoost sequentially adjusts subsequent weak learners to focus more on the previously misclassified data points. Ultimately, the combined predictions of these weak learners, weighted by their individual performance, create a strong and robust classifier.

2.5 Logistic Regression

Logistic Regression is a statistical method for binary classification, estimating the probability of a binary outcome based on one or more predictor variables. It utilizes a logistic (sigmoid) function to transform input data into a probability distribution between 0 and 1. By fitting a linear regression model to the transformed data, Logistic Regression determines the relationship between the predictor variables and the probability of the binary outcome.

2.6 Artificial Neural Network (ANN)

ANN utilizes backpropagation and gradient descent algorithms to learn intricate patterns in audio features and music genres. Through iterative training, ANNs adapt their internal parameters to optimize classification accuracy, making them powerful tools for modeling complex relationships in data and achieving high-performance genre classification in music.

3 Experiments and Results

Write about dataset, experimental setting, compare results

3.1 Dataset Description

The GTZAN dataset is a widely used benchmark dataset in the field of music genre classification. It was created by George Tzanetakis, Perry Cook, and Michael Casey for research purposes and is available on Kaggle, a popular platform for data science competitions and datasets.

3.1.1 Source

The GTZAN dataset consists of 1,000 audio tracks, each 30 seconds in duration, evenly distributed across 10 different music genres, which are:

Blues	Classical	Country	Disco	Hip-Hop
Jazz	Metal	Pop	Reggae	Rock

Table 1: Music Genres in the GTZAN dataset

3.1.2 Characteristics of the dataset

Each audio track is provided in the form of a high-quality digital audio file in .wav format, sampled at a bitrate of 16 bits and a frequency of 44.1 kHz. The audio tracks are labeled with their respective genre. The dataset is balanced, with an equal number of tracks from each genre, ensuring that classifiers are not biased towards any specific genre.

3.1.3 Relevance of the dataset

The GTZAN dataset is highly relevant and applicable for development in music genre classification algorithms. Its balanced distribution of genres and high-quality audio recordings make it suitable for training and evaluating machine learning models. Additionally, its availability on Kaggle facilitates easy access and usage.

3.2 Implementation

The six sections that the code has been divided into are as follows:

3.2.1 Importing Necessary Libraries

1. Librosa :

It is used for music and audio analysis, providing the functionality to extract music features such as Tempo, Chroma Energy Normalized, Mel Frequency Cepstral Coefficients, Spectral Centroid, Spectral Contrast, Spectral Roll off, and Zero Crossing Rate features from audio files, which are crucial for the analysis of music genres.

2. K-Nearest Neighbour Classifier :
Imported from Sci-kit Learn Library, this classifier implements the K-Nearest Neighbors algorithm, which is essential for classifying samples based on the closest training examples in the feature space.

3. Decision Tree Classifier :
Imported from Sci-kit Learn Library, predicts the value of a target variable by learning simple decision rules by answering a series of questions with either yes or no.
4. Support Vector Machine (SVC) :
Imported from Sckit Learn Library, it is used to fit to the data provided, returning a "best fit" hyperplane that divides or categorizes the data.

5. AdaBoost Classifier :

Imported from the Scikit Learn library, it is an ensemble classifier that starts by fitting a classifier on the original dataset and then adjusts weights of misclassified instances to increase the accuracy of model.

6. Logisitic Regression :
Imported from Sckit Learn Library, it is used for predictive analysis, and it is binary classification algorithm that classifies data by performing logistic regression using a cost function.

7. TenserFLow Keras :
TensorFlow.Keras provides essential tools for building neural networks. Sequential creates layered models from input to output. Dense layers build deeply connected architectures, while Dropout serves as a regularization method. The Adam optimizer adjusts the learning rate during training, and l2 regularization helps prevent overfitting by penalizing large weights.

3.2.2 Dataset Exploration I& Visualization of Different Audio Waveforms

1. Audio Waveform Visualization:
Each audio file's waveform represents the variation in air pressure for a particular sound over time. We visualized these waveforms using librosa's display.waveshow function. For example, we loaded a blues track, visually representing its waveform to observe its amplitude variations over time. This visualization helps in understanding the audio's loudness and silence patterns.

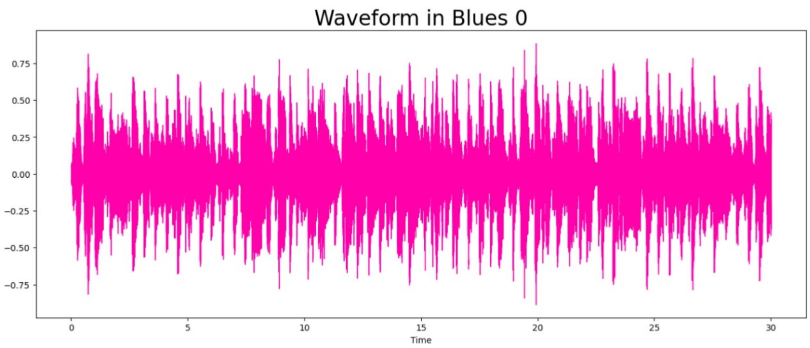

Figure 1: Waveform of Blues - 0000 song.

2. Spectrogram Analysis :
Spectrograms provide a visual way of representing the signal strength over time at various frequencies present in a waveform. We generated spectrograms using Short-Time Fourier Transform (STFT) provided by librosa.stft. The spectrogram was visualized in a logarithmic scale which is useful for identifying different musical pitches in the audio. Additionally, Mel Spectrograms were generated to focus on frequencies that are important for human hearing, which are then transformed into a Mel scale to better represent human auditory perception.

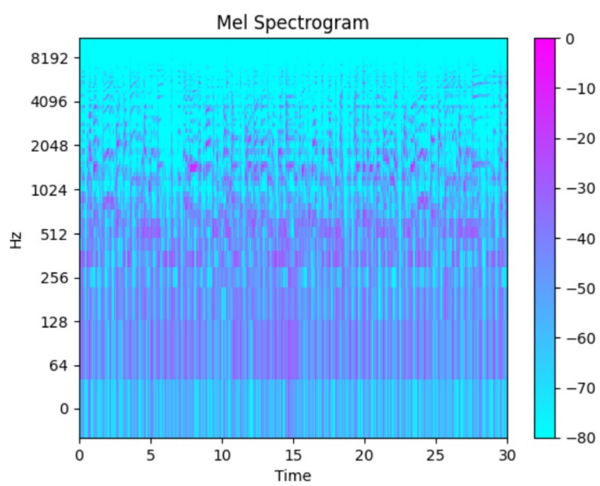

Figure 2: MEL Spectrogram

3. Chroma Feature Visualization :

Chroma features show an abstraction of the sound where only the harmonic and melodic content are captured by grouping the spectrum into 12 bins representing the 12 distinct semitones (or chroma) of the musical octave. Through this, we can observe the distinct tonal content of each genre. This was particularly useful in understanding the types of scales and chords commonly used in different musical styles.

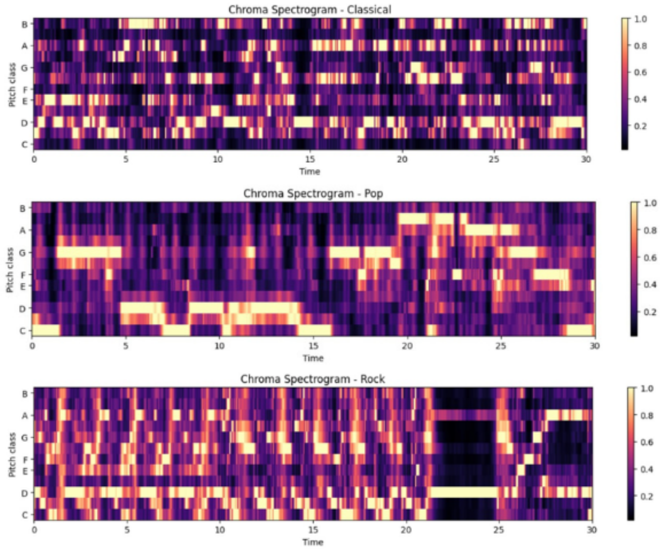

Figure 3: Chroma Spectrograms for 3 different Genres

4. Rhythm and Tempo Analysis :

To analyze the rhythmic content of the genres, we computed and visualized the tempograms that shows the local tempo over time of various audio samples using librosa's beat.tempo and feature.tempogram. This provided insights into the tempo variations within tracks, which is crucial for genre differentiation, especially for distinguishing between genres like rock and classical, which can significantly differ in rhythmic structure.

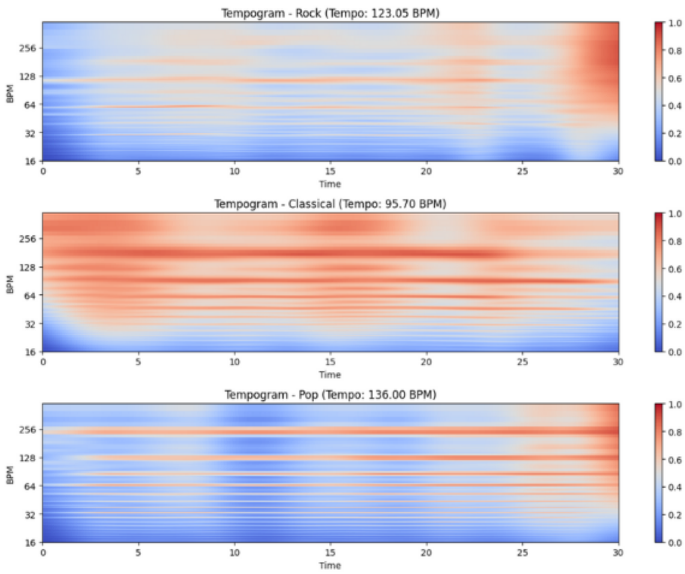

Figure 4: Tempograms for 3 different Genres

5. Harmonic and Percussive Components Separation :
Using librosa.effects.hpss, we separated the harmonic and percussive components of audio samples. Visualizing these components helped in understanding the textural differences of the tracks, where harmonic elements relate more to the tonal aspects of music and percussive elements to the rhythm.

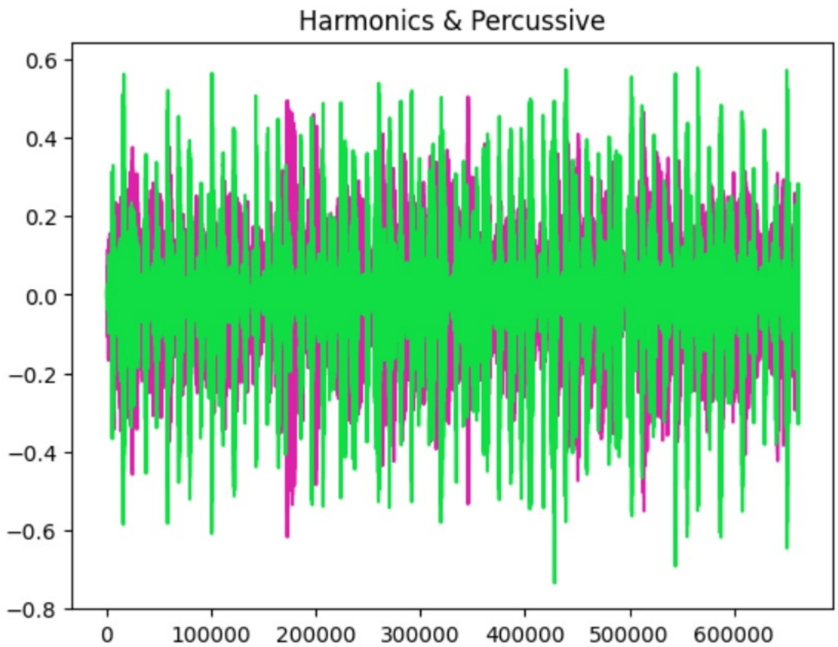

Figure 5: Harmonic and Percussive Components Separation

6. Comparative Analysis : By comparing these visualizations across different genres, we can see that classical music showed a higher prevalence of harmonic elements and a steadier tempo compared to genres like rock or pop, which displayed more percussive and dynamic rhythmic elements. This will become crucial later on this project.

3.2.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) helps in identifying patterns, and uncovering relationships between variables. In our music genre classification project, we conducted thorough EDA using the pandas, seaborn, and matplotlib libraries to gain insights into the dataset's structure and distributions.

1. Summary of Statistics:
We utilized data.head(), and data.info() to obtain a concise summary of the dataset, including the data types and any missing values. Furthermore, data.describe() was helped us to compute summary of statistics such as mean, standard deviation, minimum, maximum, and quartiles for numerical features, offering a comprehensive overview of the dataset's numerical characteristics.

2. Tempo Distribution across the genres:
To analyze the tempo distribution across different music genres, we visualized boxplots using seaborne's boxplot function. This allowed us to compare the central tendency and check for outliers in tempo values for each genre, revealing potential differences in rhythmic patterns and tempo preferences across genres. The resulting visualization provided valuable insights into the rhythmic characteristics that distinguish one genre from another.

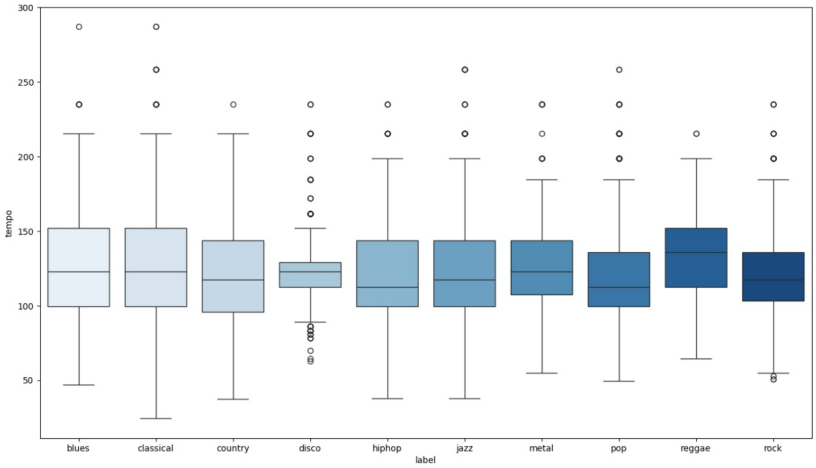

Figure 6: Box Plot for Outliers

3. Correlation Analysis:
We conducted correlation analysis to explore the relationships between the mean and variance variables extracted from the audio features. By selecting the relevant columns and computing the correlation matrix using data.corr(), we visualized the correlations using heatmaps created with seaborne's heatmap function. These heatmaps depicted the pairwise correlations between variables, highlighting any significant positive or negative correlations. This analysis helped in identifying potentially redundant or highly correlated features, informing feature selection and model building processes.

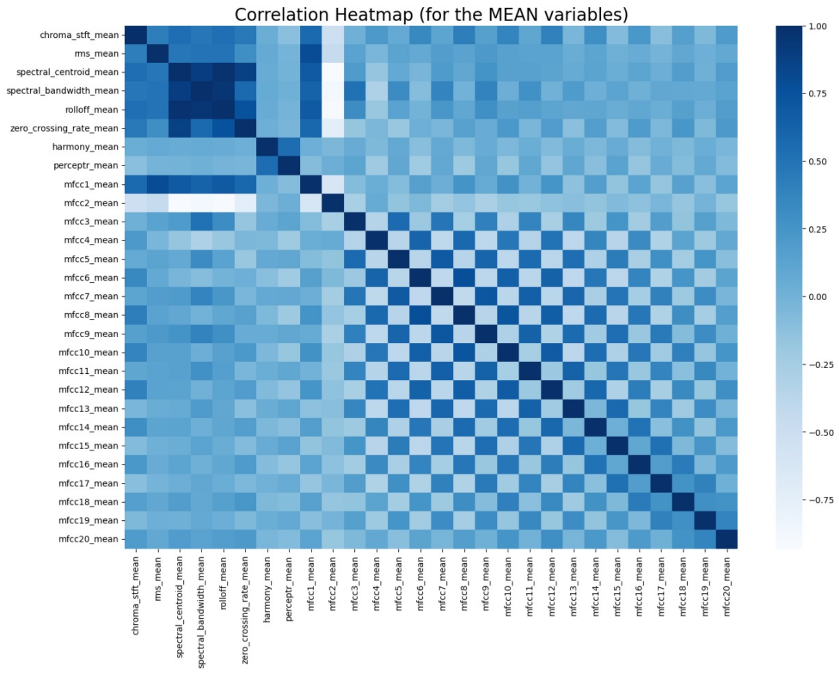

Figure 7: Correlation Heat Map for Mean Variables

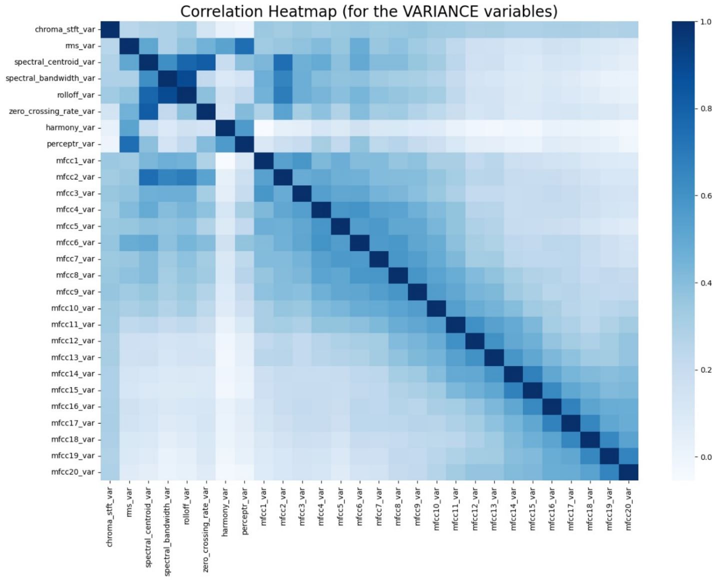

Figure 8: Correlation Heat Map for Variance Variables

4. Interesting Insights:
Through our exploratory data analysis, several key insights were revealed:

- The tempo distribution varied across different genres, with certain genres showing a wider range of tempo values than others.
- The correlation analysis provided insights into the relationships between audio features, aiding in feature selection and dimensionality reduction.
- Visualizations such as boxplots and heatmaps facilitated the identification of patterns and trends within the dataset.

3.2.4 Pre-Processing

Preprocessing plays a crucial role in preparing the dataset for machine learning model training by addressing issues such as feature scaling, dimensionality reduction, and data splitting. In our music genre classification project, we performed several preprocessing steps to ensure the dataset's quality and suitability for model training.

1. Feature Removal:
Preprocessing plays a crucial role in preparing the dataset for machine learning model training by addressing issues such as feature scaling, dimensionality reduction, and data splitting. In our music genre classification project, we performed several preprocessing steps to ensure the dataset's quality and suitability for model training.

2. Feature Scaling:
Next, we standardized the remaining features using MinMax scaling to ensure that all features have the same scale. This is important for models such as KNN and SVM, which are sensitive to feature scales. MinMax scaling transforms each feature to a specified range (typically [0, 1]), which is helpful in preserving the shape of the distribution while bringing all features to a common scale.

3. Dimensionality Reduction:
We have applied Principal Component Analysis (PCA) for dimensionality reduction, in order to enhance computational efficiency . By retaining only the most informative components while preserving most of the variance in the data, PCA reduces the dataset's dimensionality without significantly compromising its information content. We found the appropriate number of components by selecting the minimum number of principal components that explain at least 90% of the variance is equal to 24.

4. Dataset Splitting:
Finally, we split the preprocessed dataset into training and testing sets using the train-test-split function from sklearn.model-selection. Additionally, we allocated 80% of the data for training and 20% for testing, ensuring that the model is trained on a sufficiently large portion of the dataset while still having an independent dataset for evaluation.

3.2.5 Implementation of Different Models

1. K-Nearest Neighbour Classifier :
In our implementation, we utilized the grid search approach to find the optimal hyperparameters for KNN. This involved searching over a range of hyperparameters including the number of neighbors (n-neighbors), the weighting scheme (weights), and the distance metric (metric). We implemented 5 fold cross-validation to evaluate each combination of hyperparameters and selected the one that yielded the highest accuracy on the validation set. The accuracy we observed on the training data and testing data is 99.91% and 91.14% respectively

2. Decision Tree Classifier :
We employed a grid search to fine-tune the hyperparameters of the Decision Tree classifier. The hyperparameters included the maximum depth of the tree (max-depth), the minimum number of samples required to split an internal node (min-samples-split), and the minimum number of samples required to be at a leaf node (min-samples-leaf). By searching through these hyperparameters, we aimed to prevent overfitting and optimize the model's performance. The accuracy we observed on the training data and testing data is 68.62% and 58.26% respectively.

3. Support Vector Machine (SVC) :
We conducted a grid search to identify the optimal combination of hyperparameters for the SVM classifier. The hyperparameters included the regularization parameter (C), the kernel coefficient (gamma), and the type of kernel function (kernel). We aimed to find the hyperparameters that maximize the model's accuracy while avoiding overfitting, and thus get the best hyperplane. The accuracy we observed on the training data and testing data is 97.94% and 89.94% respectively.

4. Adaboost Classifier :
We utilized grid search to determine the optimal number of weak learners (n-estimators) and the learning rate (learning-rate) for the AdaBoost classifier. By iteratively training weak learners and adjusting their weights based on their performance, we can improve the accuracy of our model. The accuracy we observed on the training data and testing data is 38.95% and 38.29% respectively

5. Logisitic Regression :
We employed grid search to tune the hyperparameters of the Logistic Regression classifier, including the regularization strength (C), the optimization algorithm (solver), and the maximum number of iterations (max-iter). By selecting the optimal hyperparameters, we aimed to maximize the model's accuracy while controlling for overfitting. The accuracy we observed on the training data and testing data is 66.25% and 66.42% respectively

6. Artificial Neural Network :
We compiled the neural network using the Adam optimizer and the sparse categorical cross-entropy loss function. This step prepares the model for training by specifying the optimization algorithm and the metric used to evaluate performance. The fit method of the Sequential model was employed to train the neural network on the training data. During training, the model adjusts its weights and biases iteratively to minimize the loss function and improve its performance on the training set. We experimented with different configurations of the neural network, varying the number of neurons in each hidden layer. By adjusting the architecture of the network, we aimed to find the configuration that maximizes classification accuracy. Finally, we evaluated the trained model on the test set to assess its generalization performance. The accuracy we observed on the training data and testing data is 96.08% and 82.63% respectively

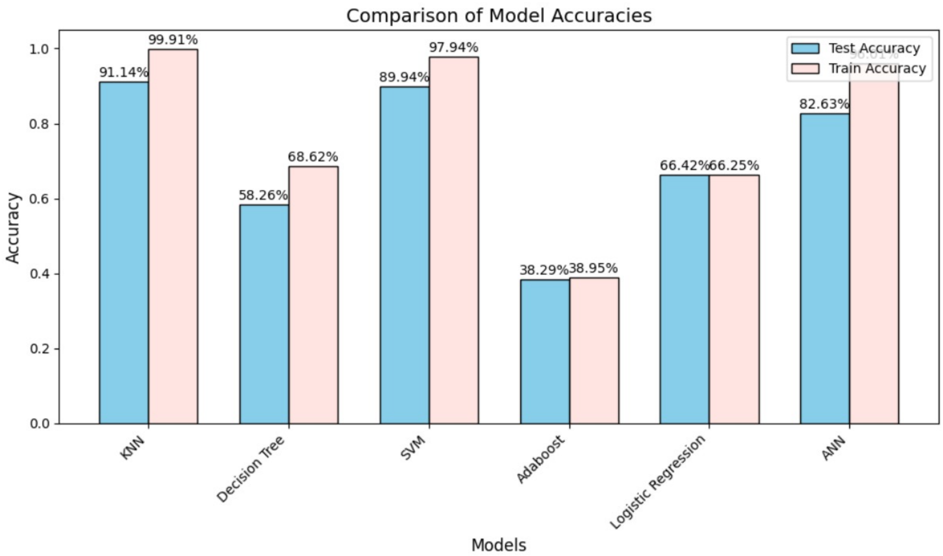

Figure 9: Comparing the accuracy of all models

3.2.6 Model Prediction

After training various machine learning models including K Nearest Neighbours (KNN), Decision Tree, Support Vector Machine (SVM), AdaBoost, Logistic Regression, and Artificial Neural Network (ANN), it's time we apply these models to predict the genres of music audio files.

1. Methodology :

- Feature Extraction: We extracted relevant audio features from the music files using the Librosa library. These features include Mel-frequency cepstral coefficients (MFCCs), chroma features, root mean square (RMS) energy, spectral centroid, spectral bandwidth, zero-crossing rate, and tempo.
- Preprocessing: The extracted features were preprocessed using techniques like scaling and principal component analysis (PCA) to standardize the data and reduce its dimensionality.
- Model Prediction: We utilized the trained machine learning models to predict the genre of each music audio file. For each model, including KNN, Decision Tree, SVM, AdaBoost, Logistic Regression, and ANN, we provided the path to the music file as input.

2. Results according various models :

We gave a music file from the blues set (blues.0000) just to check what kind of results we get from different techniques, and here are the results.

- K Nearest Neighbors (KNN): According to KNN, it is blues, which is correct.
- Decision Tree: According to Decision Tree, it is reggae, which is incorrect.
- Support Vector Machine (SVM): According to SVM, it is blues, which is correct.
- AdaBoost: According to AdaBoost, it is reggae, which is incorrect.
- Logistic Regression: According to Logistic Regression, it is blues, which is correct.
- Artificial Neural Network (ANN): According to ANN, it is blues, which is correct.

4 Summary

The project focuses on music genre classification using machine learning techniques, employing the popular GTZAN dataset containing audio recordings across ten distinct genres. The methodology involves feature extraction with Principal Component Analysis (PCA) and evaluation of various classification models such as K-Nearest Neighbors, Decision Trees, Support Vector Machines, Adaboost, Logistic Regression, and Artificial Neural Networks. To provide accessibility and interaction, a web application has been developed for model evaluation. The GitHub repository includes essential components such as the dataset, codebase for classification, pre-trained model weights, and visualization tools.

Model	**Train Accuracy**	**Test Accuracy**
KNN	99.91%	91.14%
Decision Tree	68.62%	58.26%
SVM	97.94%	89.94%
AdaBoost	38.29%	38.95%
Logistic Regression	66.42%	66.25%
ANN	96.08%	82.63%

Table 2: Final Results.