POLITECNICO
MILANO 1863

NAML Project

Classify musical genre using audio files

Peng, Rao

March 10, 2025

Table of contents

This project presents multiple machine learning models and Convolutional Neural Networks to classify music genres using audio files.

Data Exploration

- The GTZAN dataset contains 1000 audio tracks from 10 different genres.
- The dataset is divided into 10 folders, each containing 100 audio tracks from a specific genre.
- The audio tracks are in .wav format and have a sample rate of 22050 Hz.

Waveform

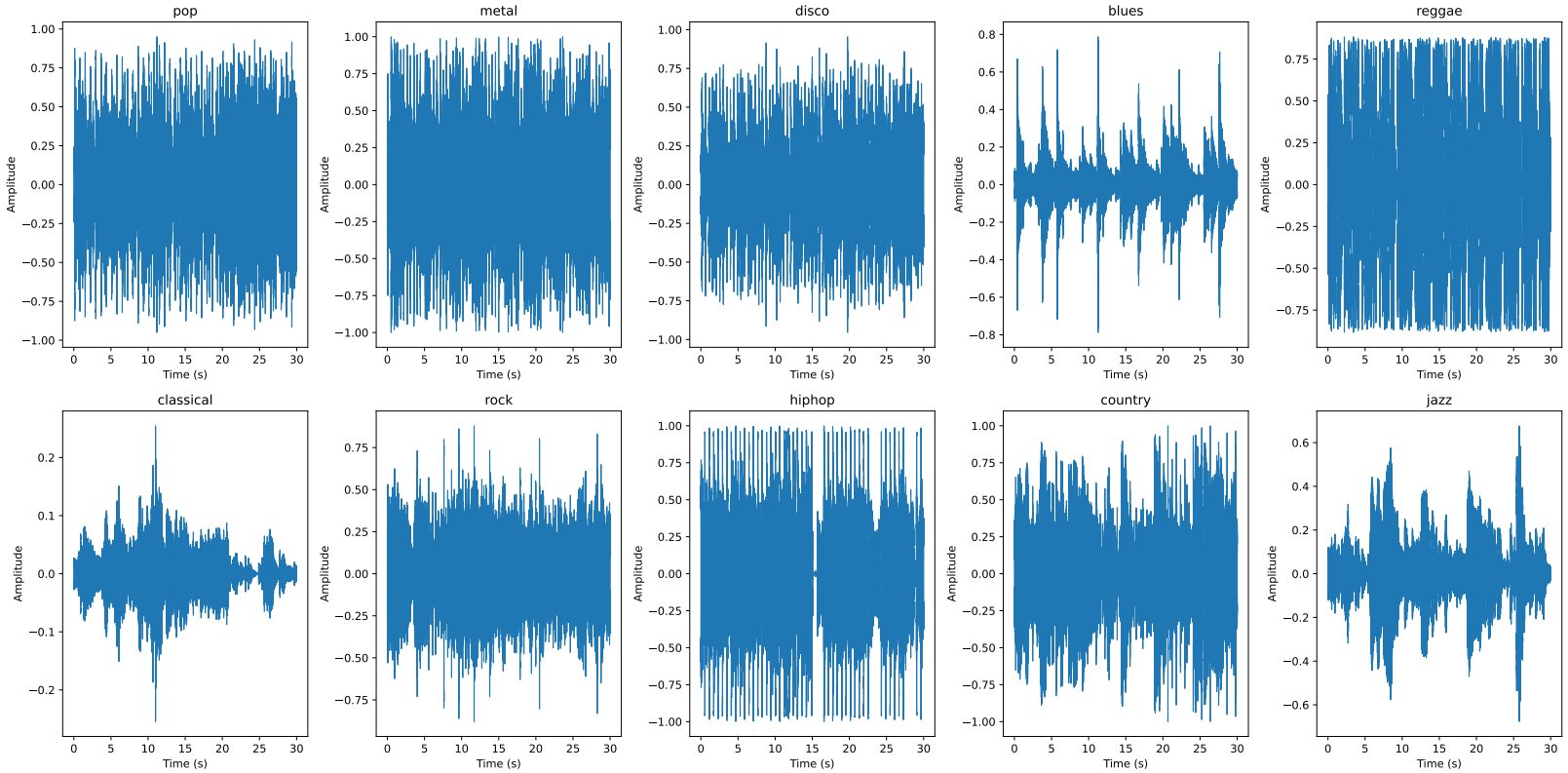


Figure: Waveform of various sample audios track over time

Chroma

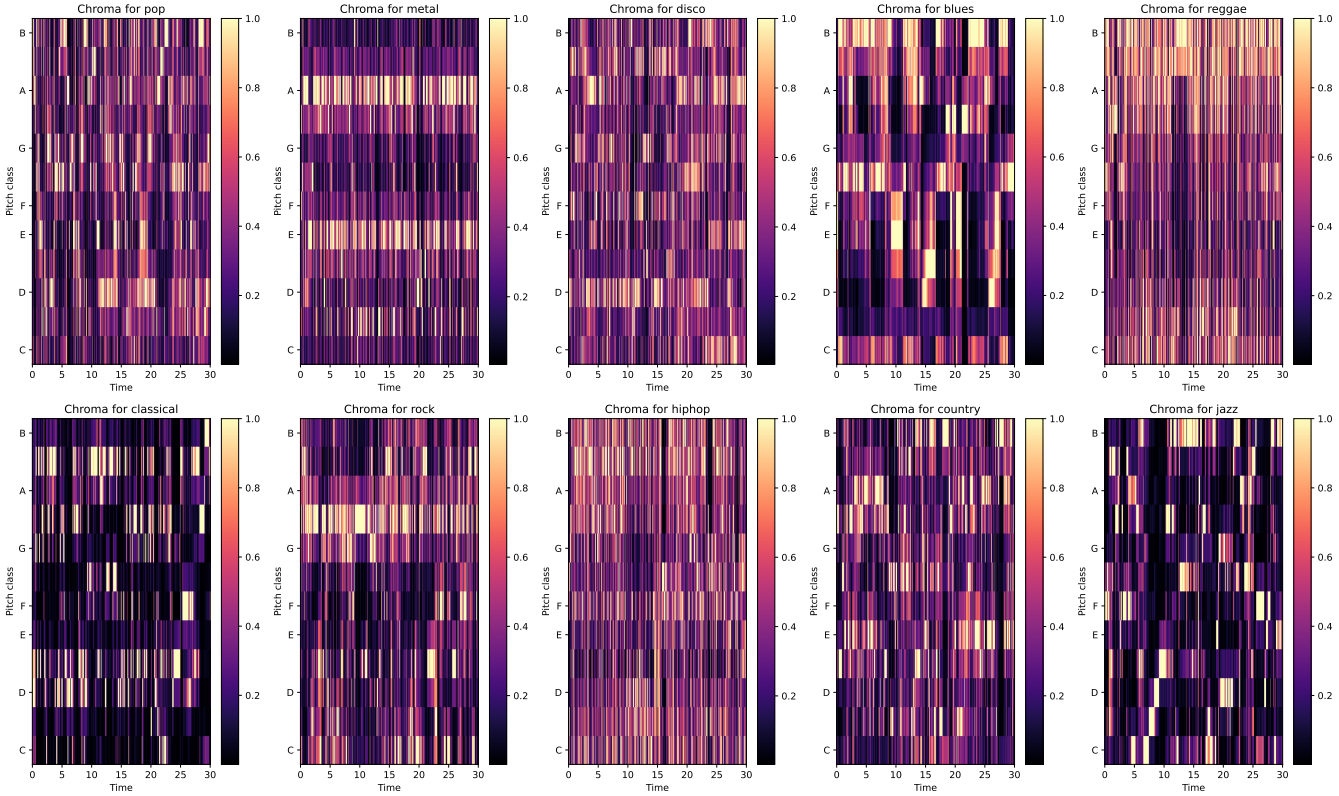


Figure: Chroma Features of various pitch classes over time

Mel Spectrogram

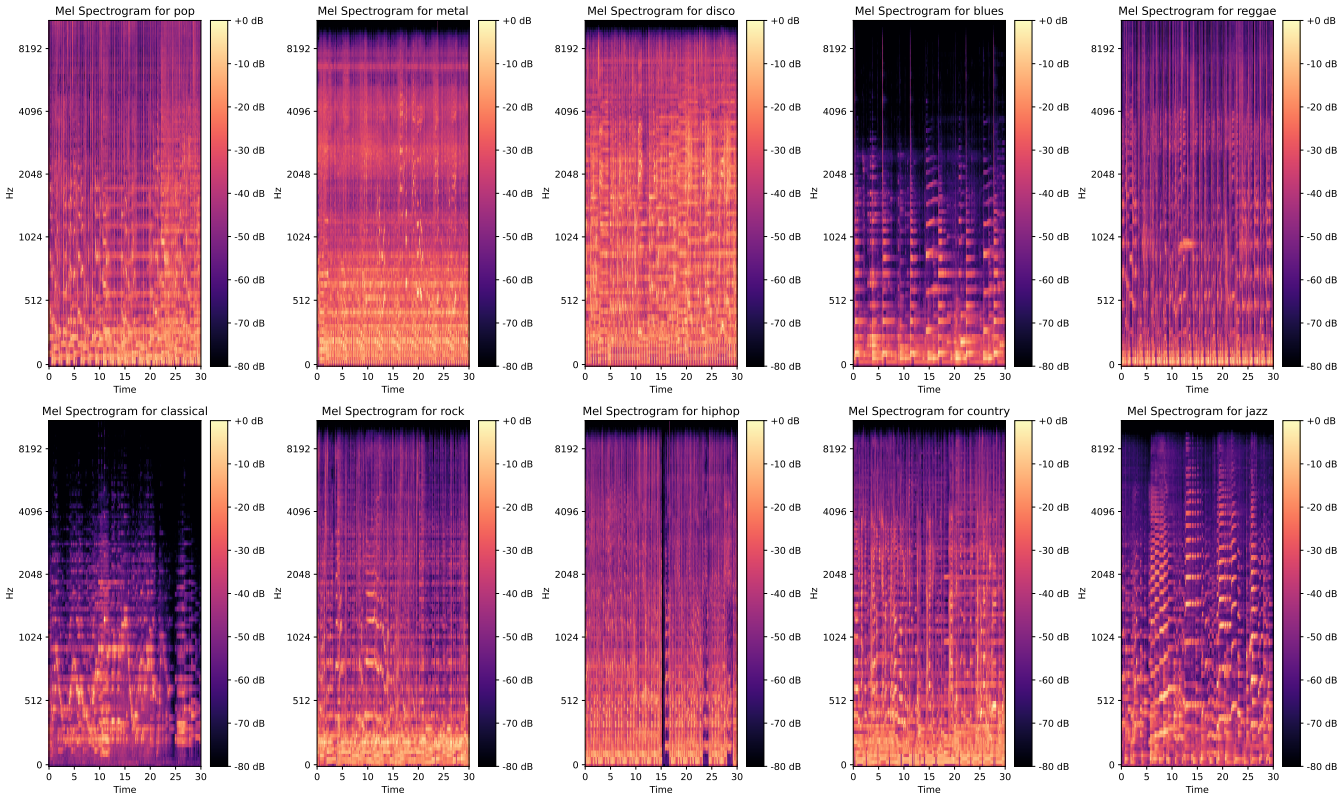


Figure: Mel Spectrogram of various sample audios track over time

Principal Component Analysis

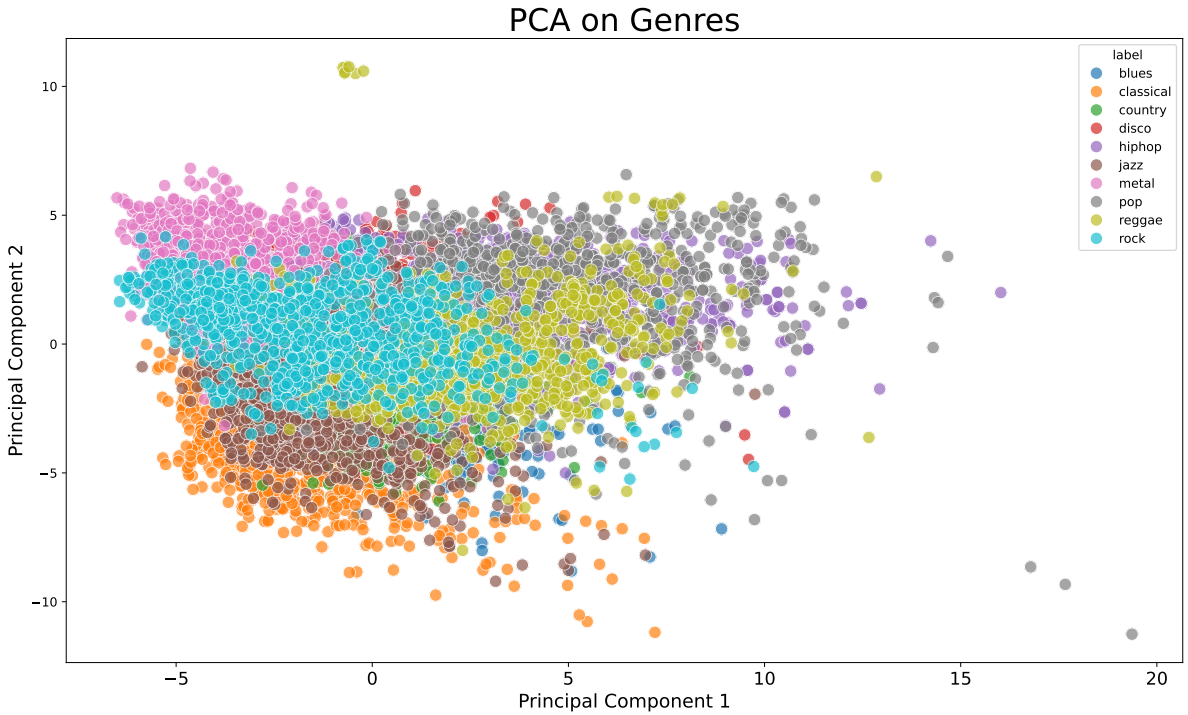


Figure: PCA of the GTZAN dataset

Normalization

I used min-max scaling to normalize the features. The formula for min-max scaling is:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Data Splitting

I split the dataset into training and testing sets using an 8/2 split. And use 5-fold cross-validation to evaluate the models.

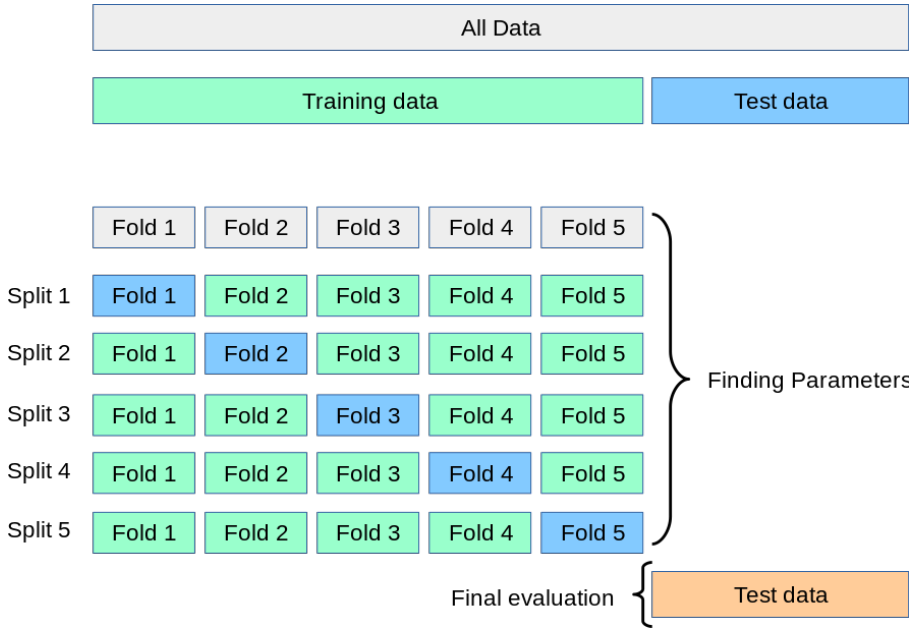


Figure: Data Splitting

Baseline Models

I trained several traditional machine learning models on the dataset to establish a baseline performance. The models I used are:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- Logistic Regression
- Random Forest
- XGBoost

Baseline Models

The results on the training set are summarized in figure, with the x-axis representing **accuracy**. The XGBoost model outperformed the other models, achieving an accuracy of 0.91 on the training set.

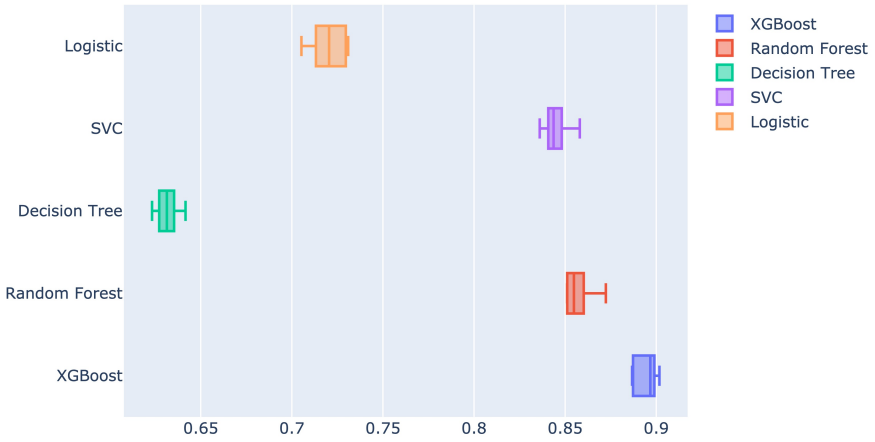
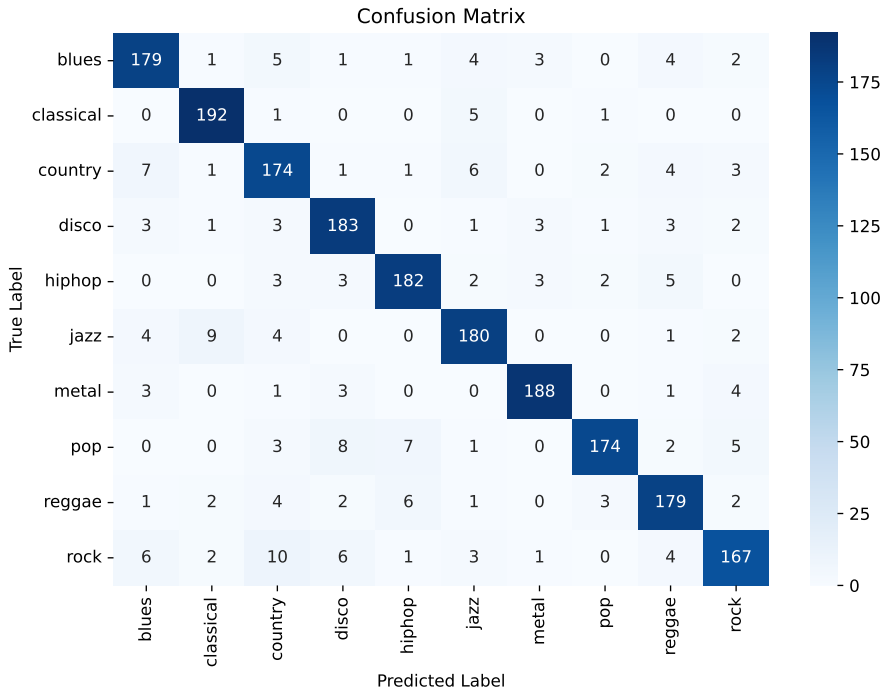


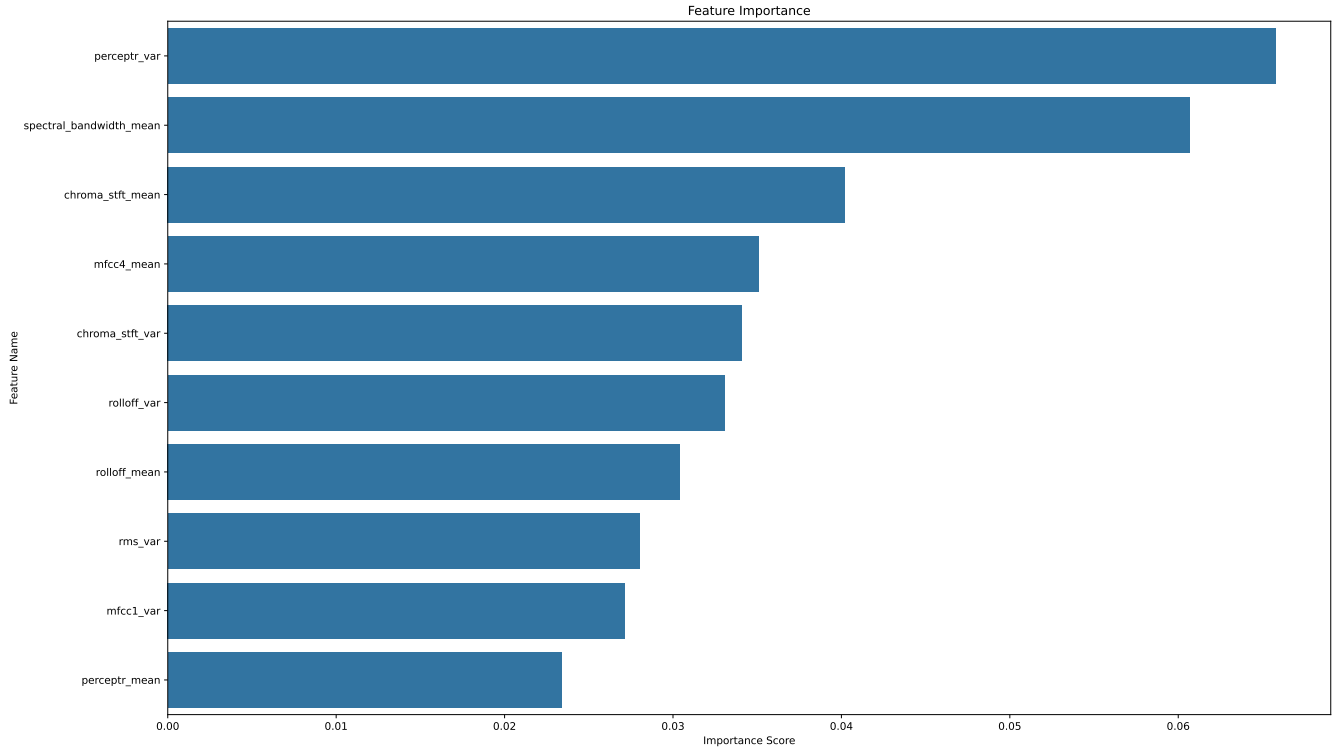
Figure: Baseline Models

Model Evaluation

I evaluated the XGBoost model on the testing set and obtained an accuracy of 0.90. The confusion matrix in figure shows that the model performs well for some genres (e.g., classical, metal) but struggles with others (e.g., rock, country).



Model Evaluation



Hyperparameter Tuning

I used **RandomizedSearchCV** to tune the hyperparameters of the XGBoost model. The best hyperparameters found by RandomizedSearchCV were:

- n_estimators: 100
- max_depth: 3
- learning_rate: 0.1
- subsample: 0.8
- colsample_bytree: 0.8

Model Architecture

The CNN architecture consists of five convolutional layers followed by max pooling layers and two fully connected layers. The model is trained on the mel spectrogram of the audio tracks.

Model Architecture

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 150, 150, 32)	320
conv2d_1 (Conv2D)	(None, 148, 148, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_3 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
dropout (Dropout)	(None, 17, 17, 128)	0
conv2d_4 (Conv2D)	(None, 15, 15, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 256)	0
dropout_1 (Dropout)	(None, 7, 7, 256)	0
conv2d_5 (Conv2D)	(None, 5, 5, 512)	1,180,160
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 512)	0
dropout_2 (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1200)	2,458,800
dropout_3 (Dropout)	(None, 1200)	0
dense_1 (Dense)	(None, 10)	12,010

Model Training

The figure illustrates a decreasing loss and increasing accuracy over time, indicating effective learning. However, as the number of epochs increases, there is a risk of **overfitting**.

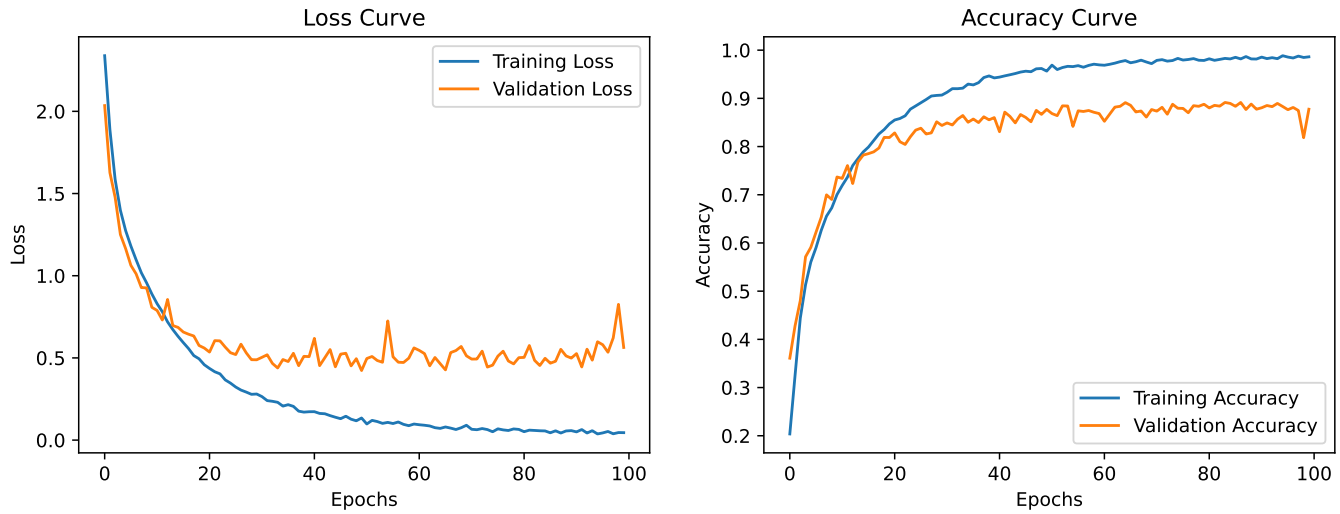
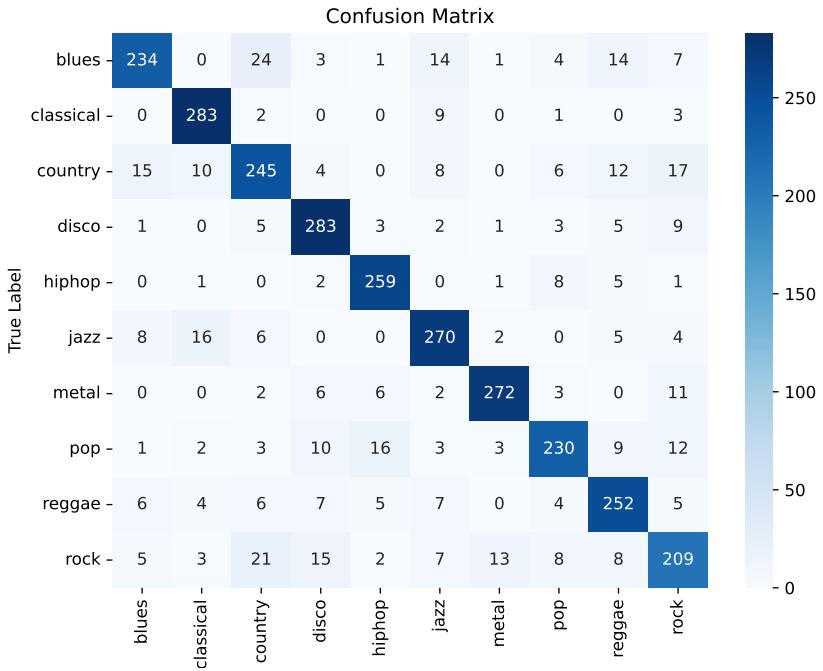


Figure: Loss and Accuracy Curves during training

Model Evaluation

It achieved an accuracy of 0.87 on the testing set. The confusion matrix in figure indicates that the CNN model performs well for most genres. However, it encounters difficulties with genres like jazz and reggae, which exhibit similar spectral characteristics.



It is evident that XGBoost is the most suitable model for this classification task. The CNN model underperformed compared to traditional models because CNNs are optimized for spatial data. Traditional models like SVC and XGBoost utilize manually extracted features (such as MFCC, Chroma, Spectral Contrast, Zero-Crossing Rate, etc.) that are specifically designed to capture audio patterns effectively.

Model	**Accuracy**	**Precision**	**Recall**	**F1score**
SVC	0.85	0.85	0.85	0.85
Random Forest	0.87	0.87	0.87	0.87
CNN	0.87	0.87	0.87	0.87
XGBoost	0.91	0.91	0.91	0.91

Table: Model Performance on testing set